$2.50

THE

SECRET GUIDE

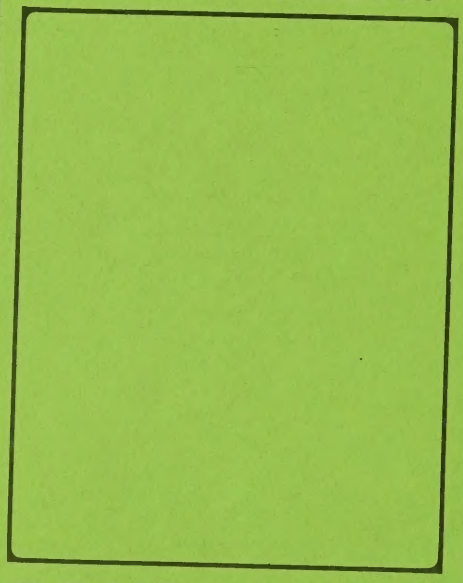
TO

COMPUTERS

PART 2: APPLICATIONS

BY RUSS WALTER, 7th EDITION, COPYRIGHT 1976.

FOR THE WIDEST SELECTION OF BOOKS AND MAGAZINES . . .



COMPUTER WAREHOUSE STORE

584 COMMONWEALTH AVE., BOSTON 02215
HOURS: M-F: 9 TO 9 • SAT: 9 TO 6 • 617/261-2701

WORDS

Russian—Can the computer translate it?—1
English—Can the computer understand it?—2

Baseball-2 Learning-2 Math-4 Encyclopedias-5 Batallions-6 Comparing words-7 Psychotherapy-7 Calculus-10 Probability-10 Paranoid schizophrenia-11

Speech—Can the computer talk?—11
Brainwaves—Can the computer read your mind?—11
Concordances—Can the computer analyze literature?—12
Authorship—Can the computer detect a ghost writer?—12
Poetry and prose—Can the computer be a poet or a novelist?—13

Balestrini-13 Ode to a bubble-14 Children's stories-14 Physics-17 Aerospace-18 A sexy tongue-19 Poetic images-20

SHAPES

Art—Can the computer be an artist?—22

John Kennedy-22 Marilyn Monroe-23 An old man-23 Pinup-23 Gulls-24 Deforming one shape into another-25 Random dots-27 Mondrian-27 Mathematical patterns-23 Famous examples-30

3-dimensional drawing—Can the computer draw what your eye sees?—32

PEOPLE

Tutors—Can the computer expand your mind?—34

Logic-34 Medicine-34 Religion-37

Personal relationships—Can the computer analyze reactions to people?—38
Robots—Can the computer do your work?—39
Bartenders—Can the computer mix your drinks?—42
Doctors—Can the computer handle your body?—42
Jobs—Can the computer give you a job?—43

EFFECTS

Jargon—Can the computer create confusing terminology?44
Artificial intelligence—Can the computer be as smart as you?—45

Hope-46 Results-52 Misinformation-55 The brain-56

Home—Can the computer invade your home?—57
Dangers—Can the computer hurt you?—63

Errors-63 Unemployment-63 Quantization-64 Asocial behavior-64 Irresponsibility-65 Concentrated power-65 Crime-65 Invaded privacy-65

Also available: Part 1 (BASIC), Part 3 (Languages), Part 4 (Systems), and the Commentary. If you'd like additional copies, drop a note to Russ Walter, 92 St. Botolph St., Apt. 3, Boston, Ma. 02116 or call (617) 266-8128.

The first computer program was written in 1949. Since then, computers have been applied to all aspects of life....

TRANSLATING RUSSIAN

Soon after computers were invented, people tried to make them translate Russian into English. I suspect Russian was chosen instead of French for three reasons:

1. Few humans could translate Russian. French translators were a-dime-a-dozen.
2. Computer experts love hard problems. Russian is harder than French.
3. Most computers were owned by the Department of Defense, which is *very* interested in Russia.

In 1954, IBM wrote a program that translated Russian sentences such as:

> Gasoline is prepared by chemical methods from crude oil. The price of crude oil is determined by the market. The quality of the crude oil is determined by the calorie content.

Unfortunately, most Russian sentences are not so simple. Recently, the end of a Russian paper on space biology was fed into a program written by Computer Concepts, Inc. Here's the translation that came out:

> Thus, the examination of some from fundamental RADIOBIOLOGICESKIX problems shows, that in this a field still very much NEREWENNYX questions. This is clear, since cosmic RADIOBIOLOGI4 is very young RAZDELOM young science efforts of scientific different specialties of the different countries of the world successful PRODOLJENY will be expanded there are.

The computer was unable to translate the words in capital letters and was stumped by Russian grammar. A competing program, written by the Air Force, translated the same passage a little better:

> Thus, consideration of from basic radio-biological problems shows that in a given region still very many unsolved questions. This and intelligibly, since space radiobiology is very young division of young science—space biology. However is base to trust that jointly scientists of different specialties of various countries of world/peace radiobiological investigations in outer space will be successfully continued and expanded.

In 1966, a special committee of the National Academy of Sciences concluded that the experience of computer translation was "uniformly discouraging" and that it was cheaper to hire a human translator than to do the two-step process of computer translation followed by human editing.

* * *

If you program the computer to translate an English sentence into Russian, and then the Russian back into English, will you get back the same English sentence you started with?

One programmer tried, "The spirit is willing, but the flesh is weak." The computer translated it into Russian, then back into English, and printed, "The booze is strong, but the meat is rotten."

Another programmer tried, "Out of sight, out of mind." The computer printed, "Blind idiot."

At an engineering conference, a computer was translating scientific papers into English, when it suddenly started talking about "water sheep." Everyone was confused. Finally they figured it out—the computer meant *hydraulic rams*.

UNDERSTANDING ENGLISH

In 1961 Bert Green, Alice Wolf, Carol Chomsky, and Kenneth Laughery at the Massachusetts Institute of Technology (MIT) programmed the computer to answer questions about baseball. In the computer's memory, they stored the month, day, place, teams, and scores of each game in the American League for one year. When the user wants to ask the computer a question, *he types the question in ordinary English.* The computer analyzes the question's grammar and prints the correct answer. Here are examples of questions the computer analyzes and answers correctly:

Who did the Red Sox lose to on July 5?
Who beat the Yankees on July 4?
How many games did the Yankees play in July?
Where did each team play in July?
In how many places did each team play in July?
Did every team play at least once in each park in each month?

To get the answer, the computer turns the questions into equations:

QUESTION	EQUATIONS
Where did the Red Sox play on July 7?	place = ? team = Red Sox month = July day = 7
What teams won 10 games in July?	team (winning) = ? game (number of) = 10 month = July
On how many days in July did eight teams play?	day (number of) = ? month = July team (number of) = 8

To do that, the computer uses this table:

WORD IN THE QUESTION	EQUATION
where	place = ?
Red Sox	team = Red Sox
July	month = July
who	team = ?
team	team =

Boston can mean either "place = Boston" or "team = Red Sox"; the computer analyzes the sentence to determine which equation to form. Words such as *the*, *did*, and *play* are ignored.

After forming the equations, the computer hunts through its memory, to find the games that solve them. If an equation says "number of", the computer counts. If an equation says "winning", the computer compares the scores of opposing teams.

* * *

In 1964, MIT awarded a Ph.D. to Bertram Raphael, for programming the computer to learn. Here are conversations between a human and the computer, using his program. The human (in *italics*) is teaching the computer how many fingers are on a hand and other information. The computer says it "understands".

There are 5 fingers on every hand.
I understand.
There are two hands on a person.
I understand.
A boy is a person.
I understand.
Tom is a boy.
I understand.
Dick is a boy.
I understand.
Harry is a boy.
I understand.
Tom has nine fingers.
I understand.
Dick has one hand.
I understand.
How many fingers does Tom have?
The answer is 9.
How many fingers does Dick have?
The answer is 5.
How many fingers does Harry have?
The answer is 10.

Every keypunch-operator is a girl.
I understand.
Any girl is an example of a person.
I understand.
Is a keypunch-operator a person?
Yes.
Is a person a girl?
Sometimes.
Is a monkey a keypunch-operator?
Insufficient information.

The man is a jerk.
I understand.
Jack is a dope.
I understand.
John is Jack.
I understand.
Is John a dope?
Yes.
Is the man a dope?
Yes.
Jim is a man.
I understand.
Is the man a dope?
Which man?

The telephone is just to the right of the book.
I understand.
The telephone is just to the left of the pad.
I understand.
Is the pad just to the right of the book?
No.
The pad is to the right of the telephone.
The above statement is already known.
The pad is to the left of the telephone.
The above statement is impossible.
The ash-tray is to the left of the book.
I understand.
The pencil is to the left of the pad.
I understand.
The paper is to the right of the telephone.
I understand.
Where is the pad?
Just to the right of the telephone.
Somewhere to the right of the pencil.
What is the position of the pad?
The left-to-right order is as follows: ash-tray, book, telephone, pad, paper. To further specify the positions, you must indicate where the pencil is with respect to the ash-tray.
The book is just to the right of the ash-tray.
I understand.
What is the position of the pad?
The left-to-right order is: pencil, ash-tray, book, telephone, pad, paper.
A telephone is an audio-transducer.
I understand.
A diaphragm is part of an audio-transducer.
I understand.
Where is a diaphragm?
Just to the left of the pad. Just to the right of the book. Somewhere to the left of the paper.

When the user begins running the program, the computer already understands words such as *a*, *the*, *is*, *every*, *has*, *owns*, *part*, and *left*. The program has two major limitations: the human's grammar must be simple, and the computer does only a few forms of reasoning.

* * *

Also in 1964, MIT awarded a Ph.D. to Daniel Bobrow, for programming the computer to solve math problems typed in ordinary English.

Let's see how the computer solves this problem:

If the number of customers Tom gets is twice the square of 20 per cent of the number of advertisements he runs, and the number of advertisements he runs is 45, what is the number of customers Tom gets?

To begin, the computer replaces *twice* by *2 times*, replaces *square of* by *square*, and replaces *per cent* by *percent*. Then the computer separates the sentence into smaller sentences:

The number of customers Tom gets is 2 times the square 20 percent of the number of advertisements he runs. The number of advertisements he runs is 45. What is the number of customers Tom gets?

The computer turns each sentence into an equation:

```
number of customers Tom gets = 2 * (.20 * number of advertisements he runs)↑2
number of advertisements he runs = 45
X = number of customers Tom gets
```

The computer solves the equations and prints the answer as a complete sentence:

```
The number of customers Tom gets is 162.
```

Here's a harder problem:

The sum of Lois's share of some money and Bob's share is $4.50. Lois's share is twice Bob's. Find Bob's and Lois's share.

Applying the same method, the computer turns the problem into these equations:

```
Lois's share of some money + Bob's share = 4.50 dollars
Lois's share = 2 * Bob's
X = Bob's
Y = Lois's share
```

The computer tries to solve the equations but fails. So it assumes "Lois's share" is the same as "Lois's share of some money", and "Bob's" is the same as "Bob's share". Now it has six equations:

```
Lois's share of some money + Bob's share = 4.50 dollars   } original equations
Lois's share = 2 * Bob's                                  }
X = Bob's                                                 }
Y = Lois's share                                          }
Lois's share = Lois's share of some money                 } assumptions
Bob's = Bob's share                                       }
```

It solves them and prints:

```
Bob's is 1.50 dollars.
Lois's share is 3 dollars.
```

Here's an even harder one:

The distance from New York to Los Angeles is 3000 miles. If the average speed of a jet plane is 600 miles per hour, find the time it takes to travel from New York to Los Angeles by jet.

The resulting equations are:

```
distance from New York to Los Angeles = 3000 * miles
average speed of jet plane = (600 * miles)/(1 * hours)
X = time it takes to travel from New York to Los Angeles by jet
```

The computer is unable to solve them. But in its memory it finds the formula "distance = speed * time". It winds up with 7 equations:

```
distance from New York to Los Angeles = 3000 * miles                  } original
average speed of jet plane = (600 * miles)/(1 * hours)                } equations
X = time it takes to travel from New York to Los Angeles by jet       }
distance = speed * time                                               formula
distance = distance from New York to Los Angeles                      }
speed = average speed of jet plane                                    } assumptions
time = time it takes to travel from New York to Los Angeles by jet    }
```

It solves them and prints:

```
The time it takes to travel from New York to Los Angeles by jet is 5 hours.
```

The computer can handle problems about age:

Mary is twice as old as Ann was when Mary was as old as Ann is now. If Mary is 24 years old, how old is Ann?

To begin, the computer replaces *twice* by *2 times*, and replaces *how old* by *what*. When the computer sees the words *as old as*, it knows the problem has to do with ages, so it breaks the problem into these sentences:

Mary's age is 2 times Ann's age X years ago. X years ago Mary's age is Ann's age now. Mary's age is 24. What is Ann's age?

The resulting equations are:

```
Mary's age = 2 * (Ann's age - X)
Mary's age - X = Ann's age
Mary's age = 24
Y = Ann's age
```

The computer prints:

Ann's age is 18.

If the situation appears hopeless, the computer asks the human for help. In this conversation, the human poses a problem, and the computer begs for a hint:

The gross weight of a ship is 20000 tons. If its net weight is 15000 tons, what is the weight of the ship's cargo?
Do you know any more relationships among these variables: gross weight of ship, tons, its net weight, weight of ship's cargo?
Yes.
Tell me.
The weight of a ship's cargo is the difference between the gross weight and the net weight.
The weight of the ship's cargo is 5000 tons.

The program has some limitations. It cannot solve quadratic equations. If the computer tries to apply the formula "distance = speed * time" to a problem involving *two* distances, it forgets which distance is which. The computer's vocabulary is limited: it doesn't realize that *how many* means *what is the number of*, and *how far is* means *what is the distance to*. If the human types *Tom has 2 apples, 3 bananas, and 4 pears*, the comma before the *and* makes the computer split the sentence into two wrong "sentences":

Tom has 2 apples, 3 bananas.
4 pears.

If the human mentions *the number of times John went to the movies*, the computer thinks *times* means multiplication, and tries to multiply *number of* by *John went to the movies*. For problems about age, the characters must be named Mary, Ann, Bill, Father, or Uncle, unless the user diddles with the computer's memory.

* * *

Also in 1964, Simmons, Klein, and McConlogue (at the Systems Development Corporation) fed a child's encyclopedia into the computer and programmed it to answer questions. If the human types, *What do worms eat?* the computer hunts through the encyclopedia, to find sentences mentioning both *worms* and *eat*. (To hasten the hunt, it uses an index produced by another program.) It finds two sentences:

Birds eat worms on the grass.
Most worms usually eat grass.

After analyzing the grammar of the question and those two sentences, the computer realizes the first sentence is irrelevant, and prints just the correct answer:

Most worms usually eat grass.

In 1965, the program's abilities were extended. If the human types *What animals live longer than men?* the computer searches through the encyclopedia for related words:

WORD IN SENTENCE	RELATED WORDS SEARCHED
animals	mammals, reptiles, fish
live	age
longer	older, ancient
men	person, people, women

Here's an example of the program's operation in 1966. The user creates some encyclopedia material about horses, and then retrieves it:

Enter: horse,noun; is,isvb; animal,noun; live,verb; on,prep; farm,noun; strong,adj; pull,verb; wagon,noun; most,adj; are,isvb; eat,verb; hay,noun; all,adj; red,adj; many,adj; have,havb; long,adj; mane,noun; some,adj; person,noun; race,verb.
Entered in dictionary.
Plurals: horse,horses; animal,animals; farm,farms; wagon,wagons; mane,manes; person,people.
Ok.
Start.
Ready.
A horse is an animal. Horses live on a farm. Strong horses pull wagons. Most horses are strong.
Learned.
Horses eat hay. All horses are red. Many horses have long manes. Some people race horses.
Learned.
Forget: all horses are red.
Forgotten.
What is a horse?
Horse is an animal.
What do horses eat?
Horses eat hay.
What do horses do?
Horses live on a farm and eat hay. Strong horses pull wagons.
What do strong horses do?
Strong horses pull wagons.
Do many horses have long manes?
Yes.
Do all horses have long manes?
To my knowledge, no.
Describe horses?
Horses are an animal. Most horses are strong. Many horses have long manes. Horses live on a farm and eat hay. Strong horses pull wagons. Some people race horses.

When the computer's answer consists of several sentences, it prints the most relevant sentence first, and the others in decreasing order of relevance.

* * *

In 1966, Thompson and Craig (at General Electric) programmed the computer to answer military questions such as:
Has the 25th batallion arrived in Texas since 3 P.M.?
Is the 100th scheduled to arrive at Ft. Lewis before the 200th leaves Ft. Lewis?

* * *

Also in 1966, Carnegie-Mellon University awarded a Ph.D. to M. Ross Quillian, for programming the computer to compare words. The user types two words; the computer finds a concept they have in common:

cry,comfort
Cry is among other things to make a SAD sound. To comfort can be to make something less SAD.
plant,man
Plant is not a ANIMAL structure. Man is ANIMAL.
To plant is for a PERSON someone to put something into earth. Man is PERSON.
plant,industry
Plant is apparatus which person use for process in INDUSTRY.

* * *

In 1966 at MIT, Joseph Weizenbaum programmed the computer to be a psychotherapist. Here's a conversation between the computer and a patient, who complains that *Men are all alike*:

Men are all alike.
In what way?
They're always bugging us about something or other.
Can you think of a specific example?
Well, my boyfriend made me come here.
Your boyfriend made you come here.
He says I'm depressed much of the time.
I am sorry to hear you are depressed.
It's true. I am unhappy.
Do you think coming here will help you not to be unhappy?
I need some help, that much seems certain.
What would it mean to you if you got some help?
Perhaps I could learn to get along with my mother.
Tell me more about your family.
My mother takes care of me.
Who else in your family takes care of you?
My father.
Your father.
You are like my father in some ways.
What resemblance do you see?
You are not very aggressive, but I think you don't want me to notice that.
What makes you think I am not very aggressive?
You don't argue with me.
Why do you think I don't argue with you?
You are afraid of me.
Does it please you to believe I am afraid of you?
My father is afraid of everybody.
What else comes to mind when you think of your father?
Bullies.
Does that have anything to do with the fact that your boyfriend made you come here?

Like the famous psychotherapist Carl Rogers, the computer asks probing questions that make the patient open up, but never gives advice: the patient must discover the truth for herself.

Instead of typing a single sentence, the patient can type a whole paragraph if she wishes. At the end of the sentence or paragraph, when she wants the computer to respond, she presses the carriage return twice.

I'll explain exactly how the program works, so you can become a psychotherapist yourself.

The computer begins by replacing some of the patient's words:

WORD THE PATIENT TYPED	REPLACEMENT
mom	*mother*
dad	*father*
dont	*don't*
cant	*can't*
wont	*won't*
dreamed	*dreamt*
dreams	*dream*

WORD THE PATIENT TYPED	REPLACEMENT
I	*you@*
me	*you*
you	*I*
my	*your*
your	*my*
myself	*yourself*
yourself	*myself*
I'm	*you're*
you're	*I'm*
am	*are@*
were	*was*

For example, the sentence *Well, my boyfriend made me come here* becomes *Well, your boyfriend made you come here.*

The computer hunts through the revised sentence or paragraph for one of these *keywords*:

CATEGORY 8: *computer,computers,machine,machines*
CATEGORY 7: *name*
CATEGORY 6: *alike,like,same*
CATEGORY 5: *remember*
CATEGORY 4: *dreamt*
CATEGORY 3: *dream,if*
CATEGORY 2: *everybody,everyone,nobody,noone,was,your*
CATEGORY 1: *always*
CATEGORY 0: *are,are@,because,can,certainly,deutsch,espanol,francais,hello,how,I, I'm,italiano,maybe,my,no,perhaps,sorry,what,when,why,yes,you@,you're*

If it finds several, it chooses the one in the highest category; if they lie in the same category, it chooses the one the patient typed first.

Then it looks up the keyword in this phrasebook:

ALIKE,SAME In what way?
ALWAYS Can you think of a specific example?
ARE *are I...:* Why are you interested in whether I am...or not? *are...:* Did you think they might not be...?
ARE@ *are you...:* Do you believe you are...? *are:* Why do you say "am"?
BECAUSE Is that the real reason?
CAN *can I...:* You believe I can..., don't you? *can you...:* Whether or not you can...depends on you more than on me.
CERTAINLY,YES You seem quite positive.
COMPUTER,COMPUTERS,MACHINE,MACHINES Do computers worry you?
DEUTSCH,ESPANOL,FRANCAIS,ITALIANO I am sorry, I speak only English.
DREAM What does that dream suggest to you?
DREAMT *you dreamt...:* Really,...?
EVERYBODY,EVERYONE,NOBODY,NOONE Really, (everybody,everyone,nobody,noone)?
HELLO How do you do—please state your problem.
HOW,WHAT,WHEN Why do you ask?
I *I remind you of:* (see *alike*) *I are...:* What makes you think I am...? *I...you:* Why do you think I...you? *I...:* We were discussing you—not me.
IF... Do you think it's likely that...?
I'M (treat as *I are*)
LIKE *(am,is,are,was)...like:* (see *alike*) *like:* (get a different keyword)
MAYBE,PERHAPS You don't seem quite certain.
MY... Why are you concerned over my...?
NAME I am not interested in names.
NO Are you saying "no" just to be negative?

REMEMBER *you remember...:* Do you often think of...? *do I remember...:* Did you think I would forget...? *remember:* (get a different keyword)
SORRY Please don't apologize.
WAS *was you...:* What if you were...? *you was:* Were you really? *was I...:* Would you like to believe I was...? *was:* (get a different keyword)
WHY *why don't I...:* Do you believe I don't...? *why can't you...:* Do you think you should be able to...?
YOU@ *you (want,need)...:* What would it mean to you if you got...? *you are...(sad,unhappy,depressed,sick):* I am sorry to hear you are (sad,unhappy, depressed,sick). *you are...(happy,elated,glad,better):* How have I helped you to be (happy,elated,glad,better)? *you was:* (see *was*) *you (feel,think,believe, wish) you...:* Do you really think so? *you (feel,think,believe,wish)...I:* (see *I*) *you are...:* Is it because you are...that you came to me? *you (can't,cannot)...:* How do you know you can't...? *you don't...:* Don't you really...? *you feel:* Tell me more about such feelings. *you...I:* Perhaps in your fantasy we...each other.
YOUR *your...(mother,father,sister,brother,wife,children):* Tell me more about your family. *your...:* Your....
YOU'RE (treat as *you@ are*)

For example, if the keyword is *sorry*, the computer looks up *sorry* in the phrasebook, which says to print "Please don't apologize."

Suppose the patient types *If the job is lousy, he'll die.* The keyword is *if*. In the phrasebook, *if* is followed by three dots, which stand for the part of the clause that comes after *if*, which is *the job is lousy*. (The computer figures out where the clause ends by looking at the punctuation.) The phrasebook says to print "Do you think it's likely that the job is lousy?"

The symbol @ serves just to locate the correct keyword in the phrasebook. Thereafter, it is ignored.

Here's what happens if the keyword is *you@*. After locating *you@* in the phrasebook, the computer ignores the @. If the patient's revised sentence contains *you want...* or *you need...*, the computer prints "What would it mean to you if you got...?" If the patient's sentence contains *you are...sad*, the computer prints "I am sorry to hear you are sad."

For each phrase in the phrasebook, Weizenbaum stored a list of alternatives. Here are the alternatives to "Please don't apologize":

Apologies are not necessary.
What feelings do you have when you apologize?
I've told you that apologies are not required.

During a conversation with a patient, the computer keeps track of which phrases it has already printed, and uses the alternatives to avoid repetition.

If what the patient types contains no keyword, the computer may print one of the following:

I am not sure I understand you fully.
Please go on.
What does that suggest to you?
Do you feel strongly about discussing such things?

Or it may take a second look at earlier parts of the conversation, retrieve a clause that contained *your...*, and print one of the following:

Let's discuss further why your....
Earlier you said your....
Does that have anything to do with the fact that your...?

For example, it may retrieve the clause *Your boyfriend made you come here* and print:

Does that have anything to do with the fact that your boyfriend made you come here?

In 1967, Weizenbaum changed the program, so it could handle keywords such as *also*, *call*, *compute*, *define*, *is*, *let*, *of*, *thanks*, *where*, and the symbol =. In the following conversation, the user teaches the computer how to find the area of a circle, a sphere, and a square, and then asks the computer some geometry questions. He also teaches the computer how to speak German—

ENGLISH	GERMAN
I say	*ich sage*
I mean	*meine ich*
what is	*wieviel ist*
squared	*quadrat*

and gradually starts speaking to the computer in German.

The area of a circle is pi times the radius of the circle squared. The area of a sphere is pi times the diameter of the sphere squared. The area of a square is the side of the square squared. The diameter of an object is twice the radius of the object. The radius of a ball is 10. What is the area of the ball?
The assumption that ball is circle is consistent—shall I go on? *no*
The assumption that ball is sphere and object is consistent—shall I go on? *yes*
It's 1256.6359.
What is the diameter of the ball?
It's 20.
The radius of a dish is 10. What's the area of the dish?
The assumption that dish is circle is consistent—shall I go on? *yes*
It's 314.15899.
The radius of a globe is 10. A globe is a sphere. A sphere is an object. What is the area of the globe?
It's 1256.6359.
The side of a plate is 10. What is the area of the plate?
The assumption that plate is square is consistent—shall I go on? *yes*
It's 100.
When I say board I mean plate. What is 3 times the side of the board?
It's 30.
Please call that x. Let f(z) be z squared minus z. Compute f(x).
It's 870.
When I say ich sage I mean I say. When ich sage meine ich I mean I mean. Wenn ich sage wieviel ist meine ich what is. Wenn ich sage quadrat meine ich squared. Nun, sagen sie mir bitte, wieviel is x quadrat.
I didn't understand you.
Ja, tut mir leid, ich sollte fragen wieviel IST x quadrat.
It's 900.
Thank you.
You're quite welcome, come again.

* * *

In 1969 at MIT, Charniak programmed the computer to solve calculus problems involving rate:

Water is flowing into a conical filter at the rate of 15 cubic inches per second. If the radius of the base of the filter is 5 inches and the altitude is 10 inches, find the rate at which the water level is rising when the volume is 100 cubic inches.
The answer is .53132943 * in * sec↑-1 * pi↑-.33333332.

* * *

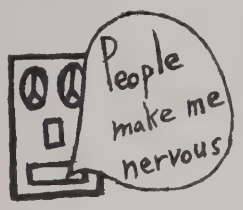
In 1971, Rensselaer Polytechnic Institute awarded a Ph.D. to J.P.Gelb, for programming the computer to solve problems about probability:

From a zorch containing a ferd and 3 brakky and 5 chartreuse werfels, 3 are drawn. What is the probability that 2 are chartreuse and the other brakky?
Replacement involved? *no*
3/22 (or 1.363636)

* * *

Also in 1971, at Stanford University, Kenneth Colby made the computer act like a paranoid schizophrenic in a mental institution. In the following conversations, psychiatrists are interrogating the computer:

How long have you been in the hospital?
About a week.
Are you depressed?
No.
Were you taking drugs?
No.
What is your problem?
People make me nervous.
How do they do that?
They stare at me.
Why?
Maybe they don't like the way I look or something.
Do you look unusual in some way?
Why are you interested in my looks?
Because you said people stare at you.
Well, sometimes.
How do you get along with people?
I don't communicate very much with the others.
Why not?
They don't realize what's going on most of the time.
What is going on?
Do you know how the underworld operates?



Tell me something about yourself.
I went to the track at Bay Meadows a while back.
What happened there?
There's a lot of crooked gambling there.
Why do you think the gambling is crooked?
I have worried that gangsters have too much power in this country.

How do you like the hospital?
I don't communicate very much with others.
Do other people at the hospital get on your nerves?
I shouldn't have come here.
Why do you say you shouldn't have come here?
They made me come here.

The computer's response depends on how much FEAR, ANGER, and MISTRUST it has at the moment. Those three variables are affected by what the psychiatrist says. For example, if the psychiatrist gives the computer a compliment, the FEAR, ANGER, and MISTRUST usually decrease. But if MISTRUST is already high, the computer interprets the compliment as a sly attempt at pacification, and its ANGER increases instead of decreases. ANGER can also be increased by questions that humiliate (*Tell me about your sexlife*) or imply subjugation (*Perhaps you should stay in the hospital longer*).

To prove the computer is paranoid, Colby had 8 psychiatrists interview it and also interview paranoid humans via teletype. Transcripts of all the interviews were sent to psychiatrists around the country, who were asked to judge whether each interview was with a human or with the computer. The psychiatrists were unable to tell the difference: only 51% of their guesses were correct.

SPEECH

Kurzweil Computer Products (in Cambridge, Mass.) has recently programmed the computer to read books to the blind. The computer is attached to a television camera aimed at the book, and also to an amplifier aimed at the blind person's ear. The computer speaks clearly, with a slightly Swedish accent.

BRAINWAVES

The computer has been programmed to read your brainwaves, and determine whether you are thinking *yes* or *no*. The program is not yet perfected: it gets the right answer 75% of the time.

CONCORDANCES

From 1957 to 1959 at Cornell University, Stephen Parrish had the computer print an alphabetical list of the words in Matthew Arnold's poetry. Here's an excerpt:

	page of book	title of poem	line of poem
CONSCIOUS			
back with the conscious thrill of shame	181	Isolation Marg	19
conscious or not of the past	287	Rugby Chapel	45
CONSCIOUSNESS			
the last spark of man's consciousness with words	429	Empedocles II	30
and keep us prisoners of our consciousness	439	Empedocles II	352
CONSECRATE			
Peter his friend with light did consecrate	445	Westmin Abbey	50
CONSECRATES			
which consecrates the ties of blood for these indeed	196	Frag Antigone	31
CONSECRATION			
won consecration from time	281	Haworth Church	46
foreshown thee in thy consecration-hour	446	Westmin Abbey	75
CONSIDER			
consider him consider not old hates	335	Merope	173
consider him consider not old hates	335	Merope	173
consider too this people who were dear	335	Merope	174

If you want to find out what Matthew Arnold said about love, just look up LOVE. Such an index is called a *concordance*. Also in 1957, John Ellison (a clergyman from Texas) published a concordance to the Revised Standard Bible; and Tasman & Busa indexed the Dead Sea Scrolls.

Joseph Raben (at Queens College) believed the poet Shelley borrowed imagery from Milton. In 1964, he had the computer make a concordance of Shelley's *Prometheus Unbound* and one of Milton's *Paradise Lost* and compare them. It found many similarities.

In 1971 at Münster University in Germany, Marvin Spevack fed the computer all the works of Shakespeare, and had it count how many times each word occurs. Disregarding trivial words such as *a* and *the*, the computer discovered Shakespeare's favorite word was *love*: he used it 2,271 times. Next come *heart*, *death*, *man*, *life*, and *hand*. He never used the word *hero*. In *Macbeth*, the word *good* occurs more often than any other adjective, noun, or adverb, and more often than most verbs.

Word counts have also been used to graph the rise and fall of themes in a novel.

DETERMINING AUTHORSHIP

In 1951, John Ellison had the computer compare 309 Greek manuscripts of the New Testament. Underneath each word of a standard text, the computer printed the variants found in the other manuscripts. It classified the manuscripts according to their similarities.

In 1963, Mosteller and Wallace had the computer determine who wrote the Federalist papers. Historians had known some of the papers were by Alexander Hamilton and others by James Madison; but the authorship of a few of the papers was in dispute. The computer compared the literary styles of the papers, by counting the frequency of words such as *by*, *enough*, *from*, *to*, *upon*, *while*, and *whilst*. It concluded that all the disputed papers were written by Madison. The statistical evidence was so high that historians accept the computer's finding as fact.

In 1964, Andrew Q. Morton (a Scottish clergyman) used the computer to deduce that Paul didn't write some of the Biblical letters attributed to him. All Morton did was count the number of times Paul used the Greek word *kai* in each sentence. *Kai* means *and*. It may seem silly to come to a conclusion about Biblical authorship by just counting the word *and*; but Morton said he analyzed 20 writers of ancient Greek and found each used *kai* with a constant frequency; in the "Pauline" letters, the frequency of *kai* varied considerably, implying some of them were not by Paul.

John Ellison distrusted Morton's assumption that a man's literary style must remain constant. He warned: if Morton's method were applied to the Declaration of Independence and Jefferson's letters to his wife, the computer might conclude that either Jefferson didn't write the Declaration of Independence or another man was writing love letters to Mrs. Jefferson. In 1965, to prove his point, he applied Morton's method to two of Morton's own articles on the subject: the computer concluded that Morton could not be the author of both!

CREATING POETRY AND PROSE

In Italy in 1961, Nanni Balestrini created poetry by feeding the computer three passages:

from Michihito Hachiya's *Hiroshima Diary*: *"The blinding fireball expands rapidly, thirty times brighter than the sun. When it reaches the stratosphere, the summit of the cloud takes on the well-known mushroom shape."*
from Paul Goldwin's *The Mystery of the Elevator*: *"...head pressed on shoulder, hair between lips, lay motionless without speaking, till he moved his fingers slowly, trying to grasp."*
from Lao Tzu's *Tao Te Ching*: *"While the multitude of things comes into being, I envisage their return. Although things flourish, they all return to their roots."*

He told the computer to mix them up. Here's the result:

TAPE MARK 1
by Nanni Balestrini and a computer

Head pressed on shoulder, thirty times
brighter than the sun I envisage their return,
until he moved his fingers slowly and while the multitude
of things comes into being, at the summit of the cloud
they all return to their roots and take on
the well-known mushroom shape endeavoring to grasp.

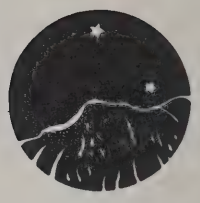
Hair between lips, they all return
to their roots, in the blinding fireball
I envisage their return, until he moves his fingers
slowly, and although things flourish
takes on the well-known mushroom shape endeavoring
to grasp while the multitude of things comes into being.

In the blinding fireball I envisage
their return when it reaches the stratosphere while the multitude
of things comes into being, head pressed
on shoulder, thirty times brighter than the sun
they all return to their roots, hair
between lips takes on the well-known mushroom shape.

They lay motionless without speaking, thirty times
brighter than the sun they all return
to their roots, head pressed on shoulder
they take on the well-known mushroom shape endeavoring
to grasp, and although things flourish
they expand rapidly, hair between lips.

While the multitude of things comes into being in the blinding
fireball, they all return
to their roots, they expand rapidly, until he moved
his fingers slowly when it reached the stratosphere
and lay motionless without speaking, thirty times
brighter than the sun endeavoring to grasp.

I envisage their return, until he moved his fingers
slowly in the blinding fireball,
they all return to their roots, hair
between lips and thirty times brighter than the sun
lay motionless without speaking, they expand
rapidly endeavoring to grasp the summit.



In Balestrini's program, *Hiroshima Diary* was represented as six phrases:

the blinding fireball
expands rapidly
thirty times brighter than the sun
when it reaches the stratosphere
the summit of the cloud
takes on the well-known mushroom shape

The Mystery of the Elevator was represented as five phrases, and *Tao Te Ching* as four. For each verse, the computer was told to choose nine phrases at random, and print them in a random order (never juxtaposing phrases from the same passage), to form six lines of roughly equal metrical length. Actually, the computer printed the poem in capital letters, without punctuation; Balestrini himself then added the punctuation and polished the grammar. The whole thing was actually done in Italian; you've been reading Edwin Morgan's translation.

* * *

A clever poetry-producing program was written at Northwestern University in 1966. The human types a list of nouns, verbs, etc. The computer randomly chooses five of the words to be *theme words*. It combines all the human's words into sentences, but chooses the theme words more often than the others. It combines the sentences into verses and tries to keep the length of each line approximately equal. It puts a theme word into the title.

In one poem, the computer chose *bubble* to be a theme word. The title was: ODE TO A BUBBLE. The poem contained phrases such as, "Ah, sweet bubble." The word *bubble* appeared so often that even the densest reader could say:

"Oh, yeah. I really understand this poem. You see, it's about a bubble."

The poem had all the familiar poetic trappings, such as "but alas!", which marked the turning-point. (Of course, you could argue that the poem didn't *really* have a turning-point, since the computer didn't have the faintest idea of what it was saying!)

* * *

In 1968 at Manchester University (England), Mendoza programmed the computer to write children's stories. Here's an example the computer produced:

The sun shone over the woods. Across the fields softly drifted the breeze, while then the clouds, which calmly floated all afternoon, moved across the fields.

Squirrel, who scampered through the trees, quickly ran off; and off noisily ran Little Grey Rabbit. She sniffed at the house; but out of the door noisily hurried Hare, who peered at slowly the flowers. Squirrel quickly scampered over the woods and fields, but Old Grey Owl flew over the woods and fields. Down the path to the woods ran Little Grey Rabbit, who then sniffed at a strawberry pie.

The first paragraph uses these words:

		VERBS						
		moved	drifted	shone	floated	touched	melted	looked-down-on	warmed
NOUNS	the clouds	1	1	0	1	0	0	1	0
	the sun	0	1	1	1	1	0	1	1
	the breeze	1	1	0	1	1	2	0	0
	the sky	0	0	0	0	1	0	1	1
ADVERBS	gently	1	1	1	1	1	1	1	1
	quietly	1	1	1	1	1	1	1	1
	loudly	1	1	1	1	1	1	1	1
	softly	1	1	1	1	1	1	1	1
	calmly	1	1	1	1	1	1	1	1
	soon	1	1	1	1	1	1	1	1
	then	1	1	1	1	1	1	1	1
	(no adverb)	2	2	2	2	2	2	2	2
ENDINGS	by	1	1	0	1				
	over the woods	1	1	1	1				
	across the meadows	1	1	1	1				
	through the trees	1	1	1	1				
	down	0	0	1	0				
	for a long time	0	0	1	1				
	all day	1	1	1	1				
	all afternoon	1	1	1	1				
	the grass					1	1	1	1
	the leaves of the trees					1	1	1	1
	the garden					1	1	1	1
	the flowers					1	1	1	1
	the little house					1	0	1	1
	the old oak tree					1	1	1	1
	the treetops					1	1	1	1

additional words: which, and, while, they, it

Mendoza stored that table inside his computer's memory. To construct a sentence, the computer randomly chooses a noun. Suppose it chooses *the sun*. Then it looks across the row marked *the sun*, to choose a verb whose score isn't 0. For example, it's possible that *the sun shone*, but not possible that *the sun melted*. Suppose it chooses *shone*. Then it looks down the column marked *shone*, to choose an adverb and an ending. Notice that the ending can't be *by*, since its score is 0. *No adverb* has a score of 2, whereas *gently* has a score of 1; that makes *no adverb* twice as likely as *gently*. If the computer chooses *no adverb* and *over the woods*, the resulting sentence is: The sun shone over the woods. In fact, that's the first sentence of the story you just read.

The computer is allowed to change the word order: instead of typing "The breeze drifted softly across the fields", the computer begins the second sentence by typing: Across the fields softly drifted the breeze. To combine short sentences into long ones, the computer uses the words at the bottom of the table: *which*, *and*, *while*, *they*, and *it*. If two consecutive clauses have the same subject, a pronoun is substituted: *they* replaces *the clouds*; *it* replaces *the sun*, *the trees*, and *the sky*. A *which* clause can modify any noun, but not a pronoun. The verb in a *which* clause must be different from the one in the main clause.

Here's the vocabulary for the second paragraph:

		VERBS							
		scampered	flew	ran	hurried	sniffed-at	peered-at	ate	munched&crunched
NOUNS	Little Grey Rabbit	0	0	2	3	1	1	0	0
	Old Grey Owl	0	3	0	0	1	3	2	2
	Squirrel	3	0	1	1	1	1	3	3
	Hare	0	0	0	2	1	1	2	2
ADVERBS	then	0	1	1	1	1	1	0	0
	slowly	0	2	0	0	1	1	1	1
	quickly	1	1	1	1	0	0	1	1
	soon	1	0	1	1	0	0	1	1
	happily	1	0	0	1	0	0	1	1
	gaily	1	0	0	1	0	0	1	1
	noisily	1	0	1	1	0	0	2	3
	(no adverb)	5	4	4	5	2	2	5	5
ENDINGS	off	1	1	1	1				
	over the woods and fields	1	1	1	1				
	through the trees	1	1	1	1				
	among the treetops	0	1	0	0				
	into the home	1	0	1	1				
	out of the door	1	0	1	1				
	down the path to the woods	1	0	1	1				
	about the garden	1	1	1	1				
	the house					1	1	0	0
	the hollow tree					1	1	0	0
	an old oak tree					1	1	0	0
	the flowers					1	1	0	0
	two buns					1	1	1	1
	a strawberry pie					1	1	1	1
	six cabbages					1	1	1	1

additional words: who, and, but, she, he

Here's another story the program produced:

The breeze drifted by. Across the fields softly moved the clouds; and then the breeze, which calmly touched the treetops, drifted across the fields. Quietly the sun shone over the woods. The sky calmly shone across the fields.

Out of the door ran Squirrel; and off hurried Hare, who munched and crunched two buns happily. Off slowly flew Old Grey Owl, and Squirrel soon ate two buns. Old Grey Owl, who peered at a strawberry pie, munched and crunched two buns; but noisily Little Grey Rabbit, who peered at an old oak tree, slowly ran down the path to the woods. Soon she hurried down the path to the woods, but then she sniffed at two buns. She hurried down the path to the woods.

Here is Mendoza's explanation of how he came to write those stories:

This work all began when a well-known scientist joined our physics department. He had spent several years away from academic life and was able to take a long cool look at academic procedures. He soon formed the theory that students never learned any ideas; all they learned was a vocabulary of okay words which they strung together in arbitrary order, relying on the fact that an examiner pressed for time would not actually read what they had written but would scan down the pages looking for these words. I set out to test his hypothesis.

I began by writing "Little Grey Rabbit" stories. I tested these stories out on my very small children; but after some minutes they grew irritable, because nothing actually happened. This shows that even small children of three can measure entropy.

Then I altered the vocabulary and grammar—making the sentences all very dead—to imitate the style of physics textbooks. The endpoint came when a colleague at another university secretly sent me an exam a week before it was given to the students. I wrote vocabularies and copied down what the computer emitted. Using a false name, I slipped my paper in among the genuine ones. Unfortunately, it was marked by a very conscientious man, who eventually stormed into the Director's office shouting, "Who the hell is this man—why did we ever admit him?" So perhaps my colleague's hypothesis was wrong, and students are a little better than we think.

Here are two of the computer's answers:

In electricity, the unit of resistance is defined by electrolysis; and the unit of charge, which was fixed at the Cavendish lab in Rayleigh's classic experiments, was measured at the Cavendish lab. Theoretically, the absolute ohm is defined in a self-consistent way. The unit of resistance, which was determined with a coil spinning in a field, was fixed at the Cavendish lab; and this, by definition, is expressed in conceptual experiments. Theoretically the absolute ohm, which was redetermined using combined e.m.u. and e.s.u., is expressed by the intensity at the center of a coil.

In this country, Soddy considered Planck's hypothesis from a new angle. Einstein 50 years ago asserted quantisation. At a photocathode, electrons which undergo collisions in the Compton effect as energy packets or quanta are emitted at definite angles; nevertheless, particles in a photocell produce photoelectrons of energy $h\nu=E_0$. Photons in vacuo transmute into lower frequencies, and light quanta in the Compton effect emit emission currents. Particles emit current proportional to energy; electrons in vacuo interact with loss of surface energy (work function); nevertheless, particles which are emitted in a photocell with conservation experimentally are conserved with energy $h\nu$. The former, at a metal surface, undergo collisions with emission of current; and at a metal surface, electrons produce emission currents. Einstein assumed the gas of quantum particles; but quite recently Rayleigh, who quite recently solved the problem in an oldfashioned way, considered radiation classically. Planck, who this century assumed the A and B coefficients, explained the gas of quantum particles but before Sommerfield; Rayleigh, who quite recently was puzzled on Boltzmann statistics, tackled the problem with disastrous results. Planck, who assumed the gas of quantum particles in 1905, this century considered the ultraviolet catastrophe; but quite recently Jeans, who tackled the problem in an oldfashioned way, was puzzled with disastrous results. Black body radiation that exerts thermodynamic forces in an engine is equivalent to a relativistic system. Out of a black body, a photon that is equivalent to (out of a black body) an assembly of photons is assumed to be a non-conservative system; at the same time, thermodynamically, black body radiation that in a piston is assumed to be a relativistic system exerts quantised forces. The radiation gas that obeys Wien's displacement law is considered as a system of energy levels. Quantally, a quantum particle exerts a Doppler-dependent pressure, although this produces equilibrium transition probabilities. Black body radiation in an engine produces equilibrium transition probabilities.

* * *

In 1968, Raymond Deffrey programmed the computer to write fake reports about the aerospace industry. I improved the program in 1973. The improved program contains these lists:

INTRODUCTORY PHRASES: in particular, on the other hand, however, similarly, as a resultant implication, in this regard, based on integral subsystem considerations, for example, thus, in respect to specific goals, utilizing the established hypotheses, moreover, in addition, in view of system operation, furthermore, to some extent, considering the postulated interrelationships, for the most part, indeed, based in system engineering concepts

NOUN PHRASES: a large portion of the interface coordination communication, a constant flow of effective information, the characterization of specific criteria, initiation of critical subsystem development, the fully integrated test program, the product configuration baseline, any associated supporting element, the incorporation of additional mission constraints, the independent function principle, a consideration fo system and/or subsystem technologies, the sophisticated hardware, the anticipated third-generation equipment, the subsystem compatibility testing, the structural design, the preliminary qualification limit, the philosophy of commonality and standardization, the evolution of specifications over a given time, the greater flight-worthiness concept, any discrete configuration made, the total system rationale

VERB PHRASES: must utilize and be functionally interwoven with, maximizes the probability of success and minimizes time for, adds explicit performance limits to, necessitates that urgent consideration be applied to, requires considerable systems analysis to arrive at, "is further compounded, when taking into account", presents extremely interesting challenges to, recognizes the importance of other systems and necessity for, effects a significant implementation to, adds overriding performance constraints to

To produce a typical sentence, the computer prints an introductory phrase, then a noun phrase, then a verb phrase, then a noun phrase. The phrases are chosen randomly.

Each paragraph consists of six such sentences. The computer is not allowed to use the same phrase twice within a paragraph. The introductory phrase is omitted from the first sentence of the first paragraph, the second sentence of the second paragraph, etc.; so the report can't begin with the word *furthermore*, and the style varies.

Here's the beginning of one such report:

THE ECONOMIC CONSIDERATIONS OF THE AEROSPACE INDUSTRY

A large portion of the interface coordination communication necessitates that urgent consideration be applied to the product configuration baseline. For example, the fully integrated test program adds explicit performance limits to the independent function principle. Moreover, the sophisticated hardware presents extremely interesting challenges to the philosophy of commonality and standardization. In view of system operation, a constant flow of effective information must utilize and be functionally interwoven with the preliminary qualification limit. In addition, any discrete configuration made adds overriding performance constraints to any associated supporting element. Thus, the anticipated third-generation equipment maximizes the probability of success and minimizes time for the total system rationale.

Moreover, the total system rationale requires considerable systems analysis to arrive at the characterization of specific criteria. Any associated supporting element necessitates that urgent consideration be applied to the structural design. In this regard, the product configuration baseline is further compounded, when taking into account initiation of critical subsystem development. On the other hand, the preliminary qualification limit adds overriding performance constraints to the greater flight-worthiness concept. In respect to specific goals, the fully integrated test program effects a significant implementation to the philosophy of commonality and standardization. In view of system operation, the independent function principle recognizes the importance of other systems and necessity for the anticipated third-generation equipment....

* * *

In 1973 I programmed the computer so that, if you give it a list of words, it will turn the words into sentence. For example, Judith Ann Abrams submitted this list:

ADJECTIVES: bitchy, creepy, dead, enthusiastic, evil, gaudy, green, gross, horny, insane, juicy, mixed-up, monstrous, multicolored, neurotic, obscene, paranoid, prickly, profane, provocative, repulsive, sensuous, sexual, sexy, shiny, sick, stimulating, strange, succulent, tasty, unsettled, weird

NAMES: Barbara, Buster, Clyde, Colin, Debbie, Ed Shanahan, Jeff, Keith, Kerri, Liz, Miles, Nicholson 700, Norwalk

NOUNS THAT AREN'T NAMES: bastard, bikini, bitch, boat, cigarette, creep, dress, ethanol, eyelid, fly, gambling, jerk, knee, kneecap, lip, mouth, navel, piece, pornography, shirt, stud, tongue, virgin, whiskey sour, whore

VERBS THAT TAKE AN OBJECT: bite, caress, castrate, climb, enjoy, kiss, knife, lick, pick, pull, rip, screw, shake, take off, teach, undo

VERBS THAT DON'T TAKE AN OBJECT: disapprove, fall down, fly, hesitate, sing, sleep, sleepwalk, sneeze, vanish, walk

EXCLAMATIONS: rats, such language, ugh

Here's what the computer printed:

A SEXY, PROFANE CREEP'S TONGUE

by Judith Ann Abrams and the computer

The lip is tasty.

Ugh!

The mouth is enthusiastic. Does the gaudy bastard sleepwalk?

The tongue's whore caresses the bikini. When Nicholson 700 is screwing the creepy jerk over a piece, Jeff is teaching the gambling. Does paranoid Liz hesitate? Why does the tongue kiss the creep stimulatingly on horny Buster?

Undo! Pull the fly repulsively! A multicolored, enthusiastic cigarette is neurotic. Is a multicolored, weird dress's navel under the strange bikini? Why is the obscene dress's cigarette Colin's ethanol? Rats! Why is the dead, insane, tasty, shiny, knifing creep's bitch unsettled and strange?

Clyde is over Ed Shanahan. Kissing the bastard, a paranoid ethanol is unsettled. Why is the prickly kneecap's mouth on the creepy piece? Teaching Norwalk is sneezing paranoidly and grossly under provocative, shiny Debbie. As a sexy, walking tongue's virgin takes off the bitchy cigarette over the profane stud then castrates enthusiastic Miles's lip paranoidly, the dead, gaudy knee is stimulating and multicolored. Even though the dead, insane piece licks strangely and neuroticly in the flying, mixed-up eyelid's knee, the dead, biting, obscene bikini is on a repulsive mouth's gambling.

The pornography is gaudy. Kerri sleepwalks. Why is the tongue sensuous? Buster is sick. Is Miles monstrous? Debbie is neurotic and paranoid, when a stimulating fly picks the navel's jerk under Ed Shanahan. Why is the dress succulent? Hesitating, a kneecap sleeps and climbs the dead, bitchy ethanol. As insane Colin's bastard falls down weirdly in a sensuous dress, green, unsettled Miles's virgin is strange and sexual.

Is the creepy eyelid provocative? The gambling's whisky sour teaches a navel.

Is the gambling evil? The bitch walks. Is the virgin profane? Why is the navel sick? Is Liz enthusiastic? Debbie enjoys the creep. Fly! Shaking, green Kerri pulls weird Colin's fly on a sick navel, then vanishes over Norwalk.

Undo a virgin! While obscene Liz is juicy and sexual, profane, gaudy Jeff's knee is under a succulent whore's navel. Tear Keith's lip bitchily and juicily on sick, weird, multicolored Barbara! Why is Buster insane? The shirt knifes the bikini. Colin shakes the bitch. The whisky sour hesitates over the green jerk. When a tasty tongue's ethanol walks, Kerri rips the boat and disapproves under enthusiastic Miles. Such language! Keith sings. Why is Buster bitchy?

Notice that the computer turned her adjectives into adverbs, by adding *ly* and making other changes. *Gross* became *grossly*, and *juicy* became *juicily*. This approach was not always successful: *stimulating* became *stimulatingly* (a non-existent word), and *neurotic* became *neuroticly* (instead of *neurotically*).

It conjugated her verbs. *Screw* became *screwing*, and *bite* became *biting* (the computer dropped the *e*). *Lick* became *licks*, and *teach* became *teaches* (the computer added the *e* after the *ch*).

It added *'s* to her nouns. *Jeff* became *Jeff's*. *Miles* became *Miles's* (it should have become *Miles'*).

The grammar is chosen randomly. For each sentence, there is a 10% it will begin with an exclamation. If the sentence is not merely an exclamation, there is an 18% chance it will be a question.

If it is a question, there is a 40% chance it will be preceded by the word *why*. There is a 50% chance the main part of the question will have the form *does*...noun phrase...verb phrase, and a 50% chance it will have the form *is*...noun phrase...complement.

The definitions of *noun phrase* and *verb phrase* and *complement* involve random numbers also.

There is a special variable, called W. At the beginning of the composition, W is near zero; but it tends to increase as the composition progresses. It affects the *complexity*. When W is large, the chance of printing adjectives, adverbs, subordinate clauses, and correlative clauses is large. This sentence was produced by a small W: *The lip is tasty.* This sentence was produced by a large W: *As a sexy, walking tongue's virgin takes off the bitchy cigarette over the profane stud then castrates enthusiastic Miles's lip paranoidly, the dead, gaudy knee is stimulating and multicolored.*

* * *

Also in 1973, one of my students (Toby D'Oench) made the computer produce poetic images, such as these:

TO GUINEVERE—LADY OF THE LAKE

Silent mists
Billow in creations
Windmills for flames evolve into ethers
Merlin again

MY MEMORY

Frozen children
Quiver with leaves
Creations with leaves hover over thoughts
Gardens of verse

A NEW ENGLAND BARN

Lazy fragrances
Waft by ethers
Seas on fragrances billow in sorrow
Rusted pitchforks

NEWPORT

Frozen sails
Slumber in fog
Hazes for sails waft by thoughts
Docks—yachts—luxuries of eras gone by

The program contains these lists:

TITLE...NOUN...ENDING

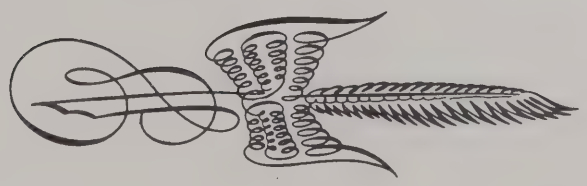
TO REMBRANDT...windmills...A simple brush
WAITING FOR THE PATIENT...ethers...Waiting
THE PROPHET...visions...Then a word
LISTERINE...breaths...Plastic society
EINSTEIN...thoughts...Somehow through this—an understanding of a superior order
NEWPORT...sails...Docks—yachts—luxuries of eras gone by
EXISTENCE...seas...In the beginning?
SUMMER IN WATTS...flames...Tar-street neon—and the night
TO GUINEVERE—LADY OF THE LAKE...mists...Merlin again
NOON IN CALCUTTA...hazes...Emaciated dark forms strewn like garbage
WEST HARBOR...fog...A solitary gull slices through
A NEW ENGLAND BARN...fragrances...Rusted pitchforks
A CHILD'S MICROSCOPE...creations...The wonderful amoeba
A GROUP PORTRAIT...bundles...Christmas
THE MILKY WAY...cosmos...A gooey mess
TOMBSTONE...sorrow...Rubbings
LIFE AT THE END OF A BRANCH...leaves...Swirling to the ground
SEASHELLS AND THINGS...waves...Dribble-dribble-dribble castle
A BEAVER POND...reeds...Thwack
MY MEMORY...children...Gardens of verse

ADJECTIVES: fleeting,crimson,silent,sensate,pliant,gloomy,pallid,inky,frozen,lazy
PREPOSITIONS: of,on,under,above,below,in,with,by,for,through
VERBS: billow in, glitter with, flutter by, drift with, flow into, ponder about, waft by, quiver with, hover over, gleam like, wander through, slumber in, dart by, evolve into, sing to

Each poetic image has this form:

TITLE

adjective noun that goes with the title
verb noun
noun preposition noun verb noun
ending that goes with the title

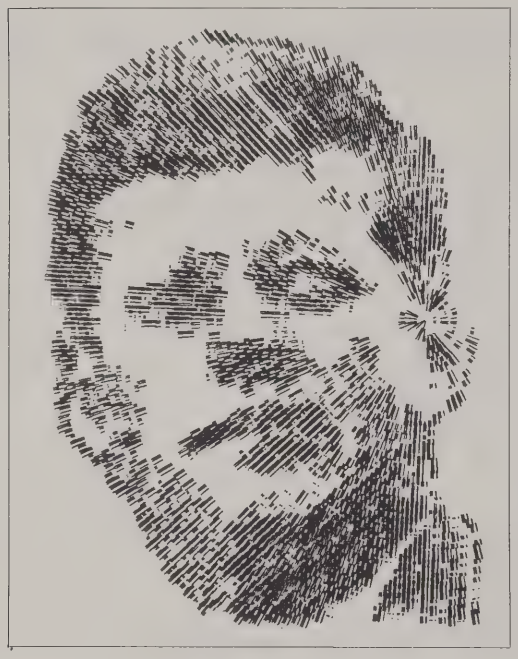


CREATING ART

Every black-and-white photograph can be expressed as a table of numbers. Each number in the table represents the darkness of a different point—the higher the number, the darker the point. The "darkness numbers" are called *gray levels*. To feed a picture into the computer, type in the table of gray levels. Or aim a special camera (called an *optical scanner*) at the object you want pictured; the scanner will automatically compute the gray levels and send them to the computer via a wire.

You can program the computer to change the gray levels in any weird way you wish, and draw the result. Here's what the Computer Technique Group of Japan did to an ordinary photograph of John Kennedy:

SHOT KENNEDY

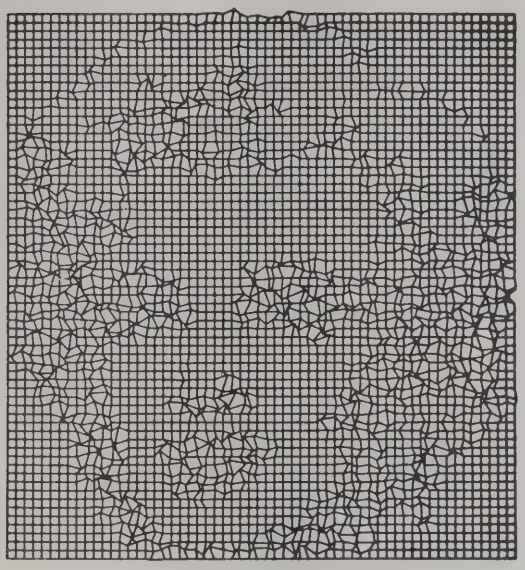


DIFFUSED KENNEDY

KENNEDY IN A DOG

Here's what the group did to a photograph of Marilyn Monroe:

MONROE IN THE NET

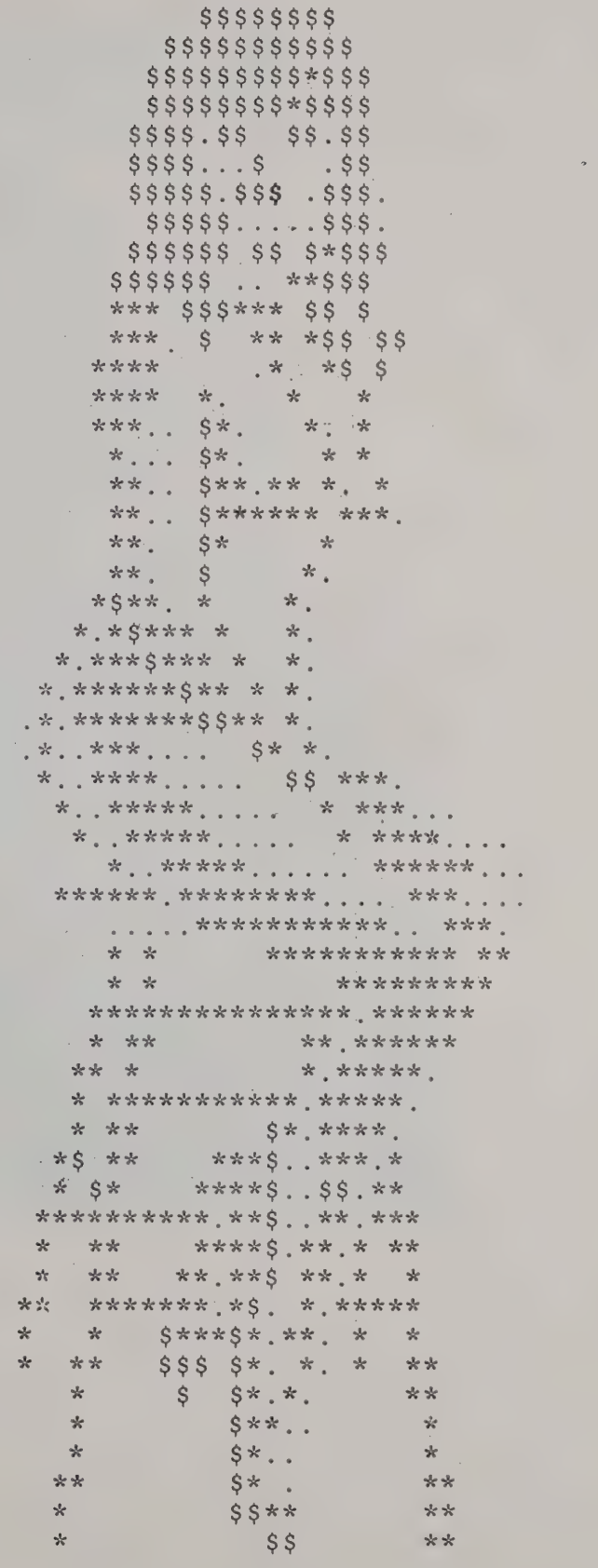


Csuri & Shaffer fed the computer a realistic line drawing of an old man; here's what came out:

RANDOM LIGHT AND SHADOW

I did this with the help of a computer:

PIN-UP



The *Pin-Up* has these specifications....

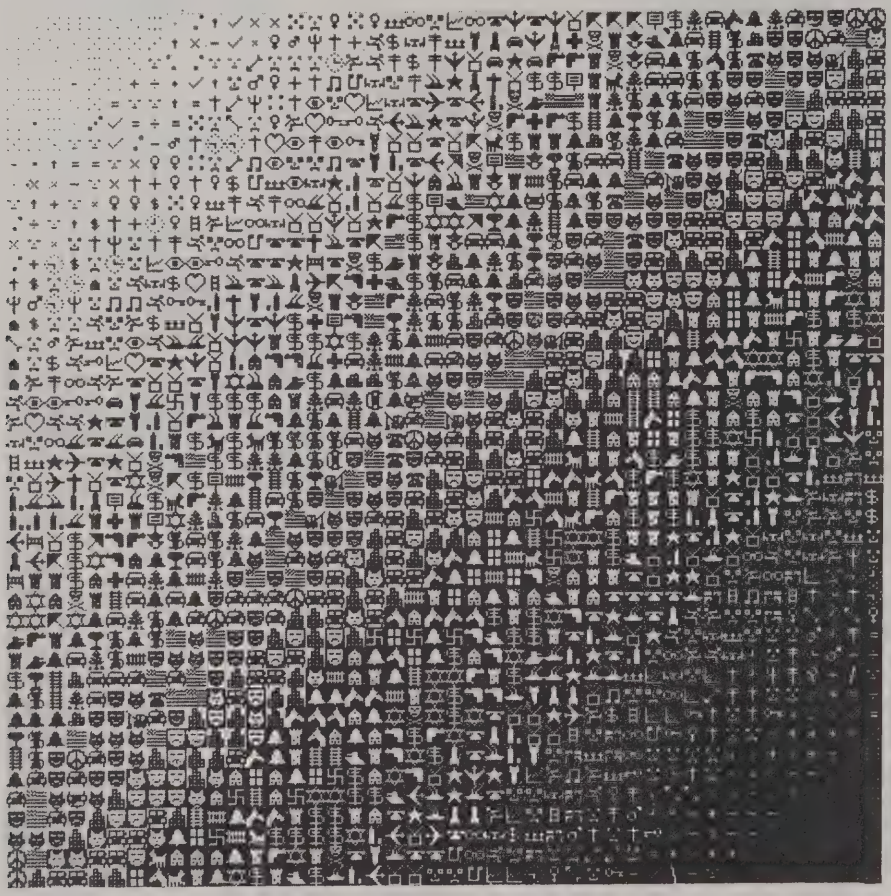
scene: a scantily clad girl sitting on a stool
4 different gray levels
4 different symbols (1 for each gray level): a blank, a period, an asterisk, and $
1537 symbols altogether (53 rows × 29 columns)

In the specifications, the numbers are small, yet the picture is clear. The clarity was obtained by non-computerized fanagling.

At Bell Telephone Laboratories, Knowlton & Harmon produced a picture with much larger specifications....

scene: two seagulls flying in the clouds
16 different gray levels
141 different symbols (for each gray level there are several symbols; the computer chooses among them at random)
11616 symbols altogether (88 rows × 132 columns)

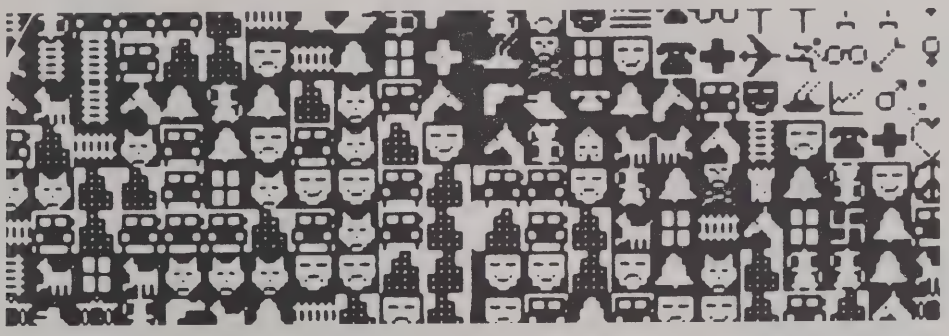
Instead of using blanks, periods, asterisks, and $, they used cats, battleships, swastikas, and other weird shapes. Here are the 141, listed from lightest to darkest, with some repetitions:



The picture is several feet long. Seen from a distance, it looks like this:

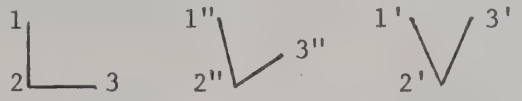
GULLS

Here's a close-up view of part of one of the gull's wings:



Here's how to make an L slowly become a V. Notice that the letters L and V are both made by connecting three points:

Let 1" be the point halfway between 1 and 1'; let 2" be halfway between 2 and 2'; and let 3" be halfway between 3 and 3'. Then 1", 2", and 3" form a shape that's halfway between an L and a V:



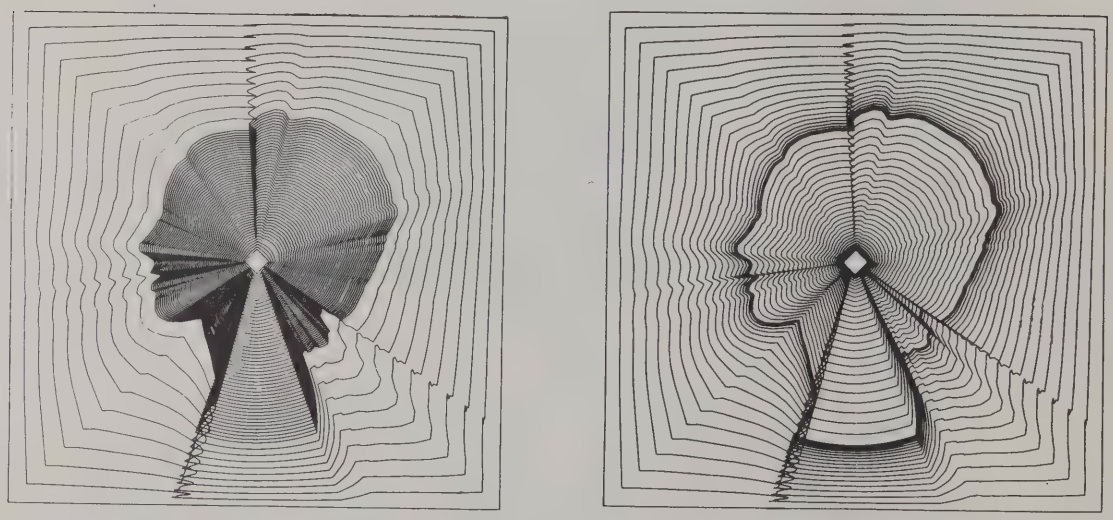
The process can be extended further:

Using that method, the Computer Technique Group gradually turned a running man into a Coke bottle, and then into Africa:

RUNNING COLA IS AFRICA

The group turned this head into a square:

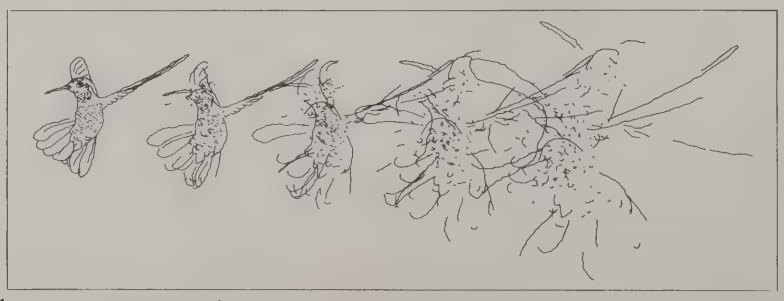
RETURN TO A SQUARE



The head on the left returns to a square by using *arithmetic progression*: the lines are equally spaced. The one on the right uses *geometric progression* instead: the lines are close together near the inside square, but further apart as they expand outward.

Csuri & Shaffer exploded a hummingbird:

CHAOS TO ORDER



The hummingbird at the far right was obtained from the one at the far left, by moving each line a random distance and in a random direction (between 45° and -45°).

Computer artists are starting to believe that art is a tension between order and disorder. Too much order, or too much disorder, is boring. For example, in *Chaos to Order*, the hummingbird on the left is too orderly to be art. The hummingbird on the right is more interesting.

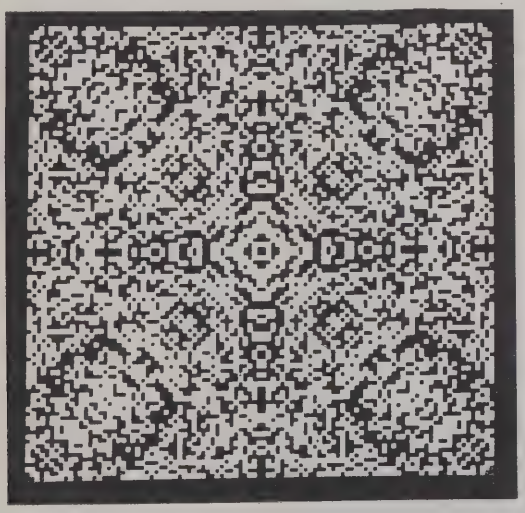
Consider *Gulls* (page 25). Seen from a distance, it is an orderly picture of gulls. Seen up close, it is an orderly picture of a cat or a battleship or a swastika. But from a middling distance, it looks like disorderly wallpaper—the symbols repeat, but not in any obvious cycle. That element of disorder is what makes the picture interesting.

At first glance, *Pin-Up* (page 23) is just a disorderly array of periods, asterisks, and dollar signs. At second glance, you see order: a girl. Art is the formation of order from disorder.

A first glance at *Monroe in the Net* (page 23) shows order: a piece of graph paper. A second glance shows disorder: some of the graph's lines are inexplicably bent. A third glance shows order: Marilyn Monroe's face pops out at you. Her orderly face is formed from the disorder of bent lines.

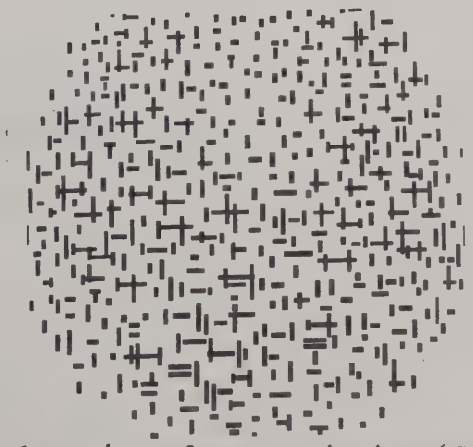
Return to a Square (page 26) uses arithmetic progression and geometric progression to create an over-all sense of order, but the basic elements are *dis*orderly: a head that's bumpy, and a panorama of weird shapes that lie uncomfortably between being heads and squares but are neither.

Many programs create disorder by random numbers. *Chaos to Order* uses random numbers to explode the hummingbird. *Gulls* uses random numbers to help choose among the 141 symbols. An amazing example of random numbers is this picture by Julesz & Bosche:



To your eyes, the picture seems quite ordered. Actually, it is quite *dis*ordered. One pie-shaped eighth of it is entirely random; the other seven eighths are copies of it. The copying is the only element of order, but very powerful. Try this experiment: *cover seven eighths of the picture*. You'll see that the remaining eighth is totally disordered, hence boring.

Take this test:



One of these is a famous painting (*Composition with Lines*, by Piet Mondrian, 1917). The other was done by a computer (programmed by A. Michael Noll in 1965). *Which one was done by the computer? Which one do you like best?*

The solution is on the next page, but don't peek until you've answered.

The computer did the top one.

The programmer surveyed 100 people. Most of them (59) thought the computer did the bottom one. Most of them (72) preferred the top one—the one that was actually done by the computer.

The test shows that people can't distinguish computer art from human art, and that the computer's art is more pleasing than the art of a famous painter.

The version produced by the computer is more disordered than Mondrian's. The computer created the disorder by using random numbers. The survey shows that most people like the disorder: Mondrian's work is too ordered. It also shows that most people mistakenly associate *computer* with *order*.

If you feed the computer these instructions, here's what it will draw:

---

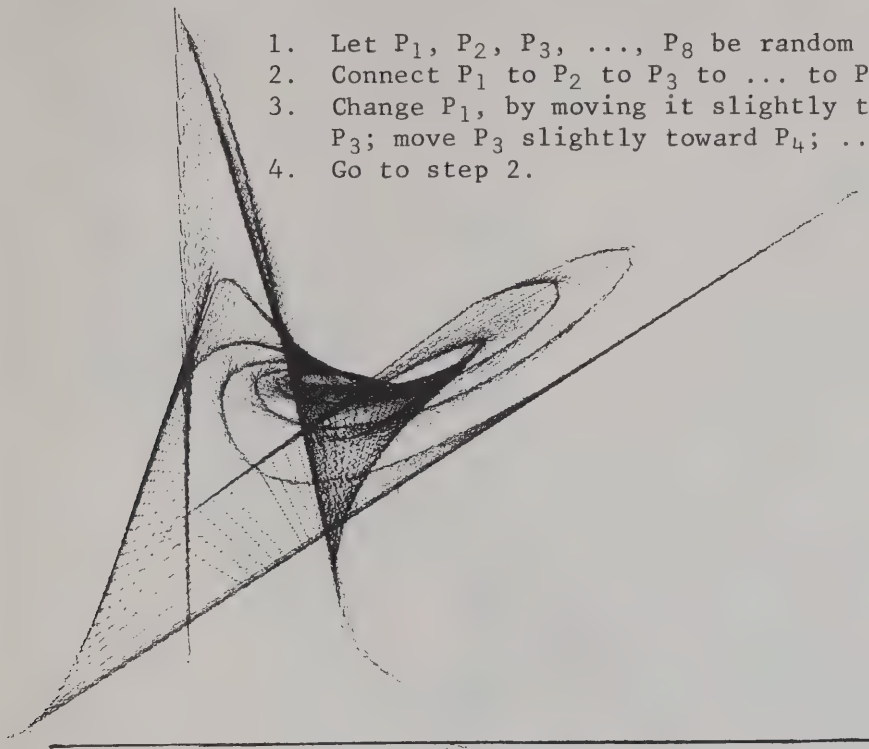
1. Let P_1, P_2, P_3, ..., P_{23} be equally-spaced points on a circle.
2. Connect each point to each of the other 22 points.

---

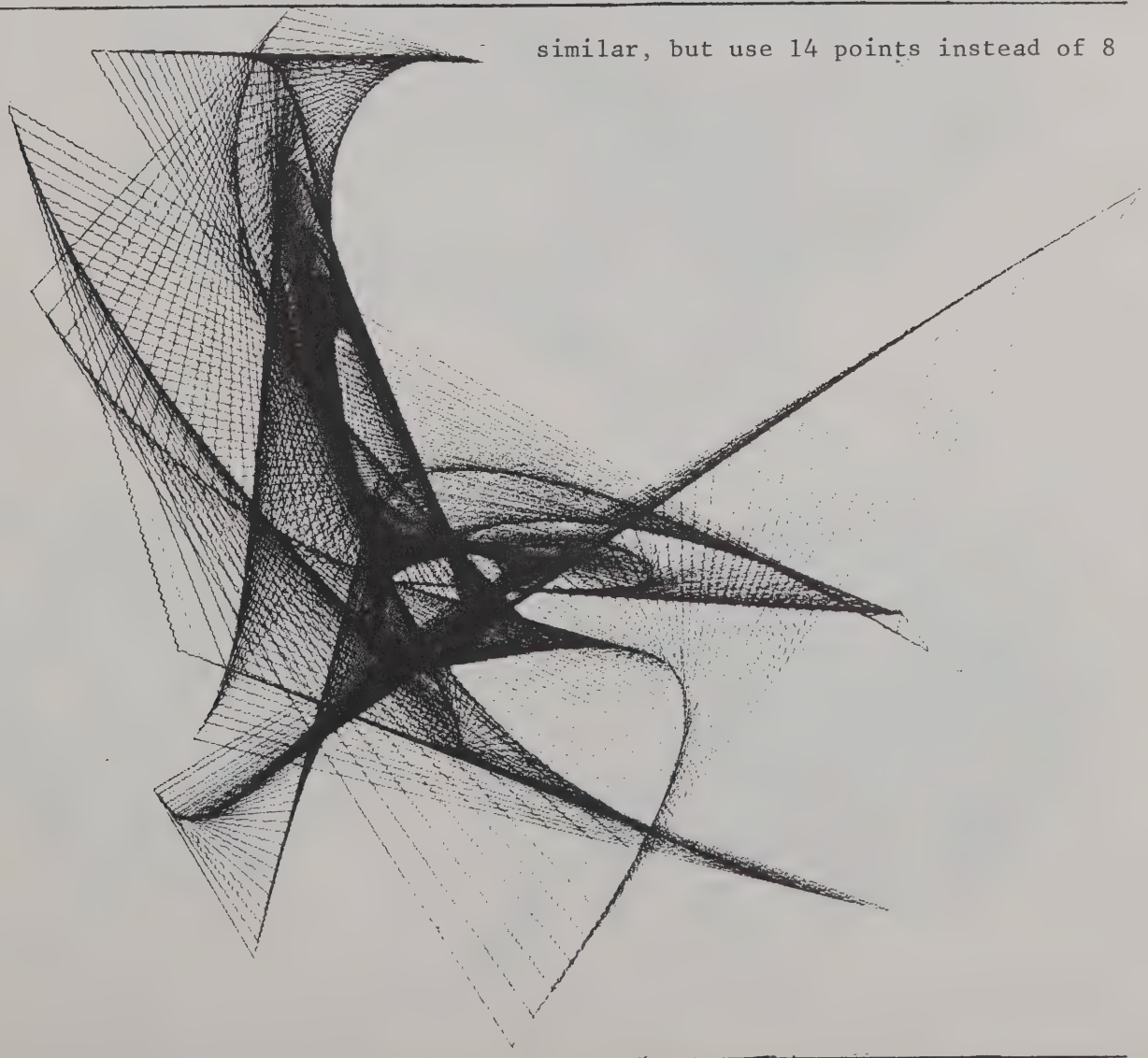
similar, but use 24 points instead of 23

---

1. Let P_1, P_2, P_3, ..., P_8 be random points.
2. Connect P_1 to P_2 to P_3 to ... to P_8 to P_1.
3. Change P_1, by moving it slightly toward P_2; move P_2 slightly toward P_3; move P_3 slightly toward P_4; ...; move P_8 slightly toward P_1.
4. Go to step 2.

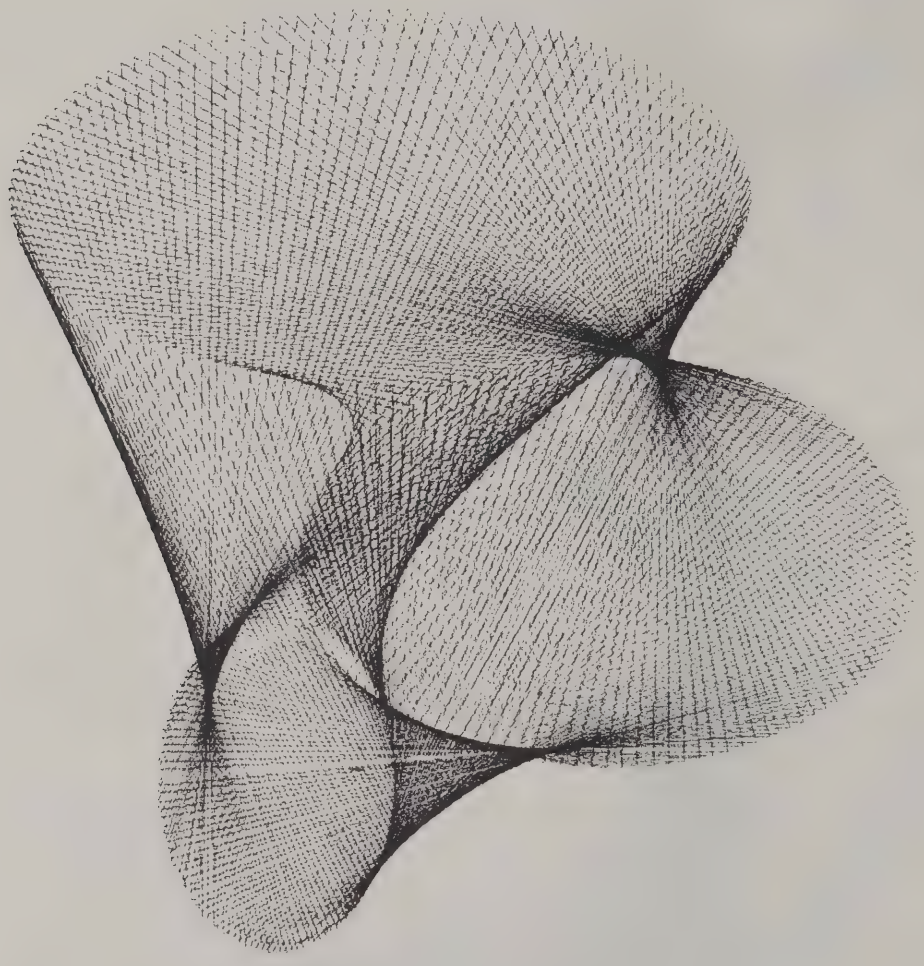


similar, but use 14 points instead of 8



programmed by Toby D'Oench (one of my students)

1. Let P, Q, and R be ellipses, whose sizes and locations depend on random numbers.
2. Let P_0, P_1, P_2, ..., P_{125} be equally-spaced points on ellipse P; let Q_0, Q_1, Q_2, ..., Q_{125} be equally spaced on Q; let R_0, R_1, R_2, ..., R_{125} be equally spaced on R. On each ellipse, label the points counterclockwise, and so that P_0, Q_{42}, and R_{84} are the rightmost points.
3. Draw a triangle whose vertices are P_0, Q_0, and R_0; draw one whose vertices are P_1, Q_1, and R_1; ...; draw one whose vertices are P_{125}, Q_{125}, and R_{125}.



Here are some famous examples of computer art:

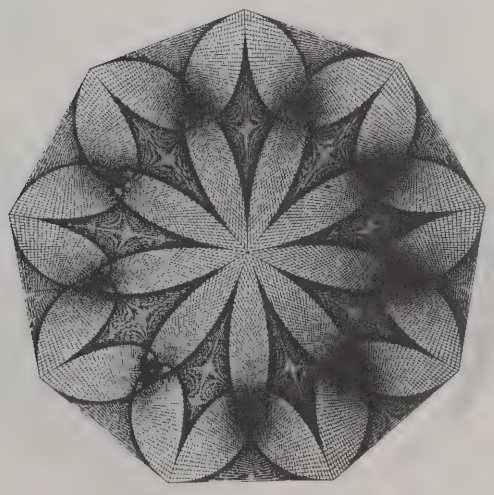
CUBIC INTERPOLATION
by Schmitt & Gaines

HUMMINGBIRD
by Kerry Strand

THE SNAIL
by Kerry Strand

KRYSTALLOS
by Strand & Jenkins

SYMPLEXITY
by Strand & Jenkins



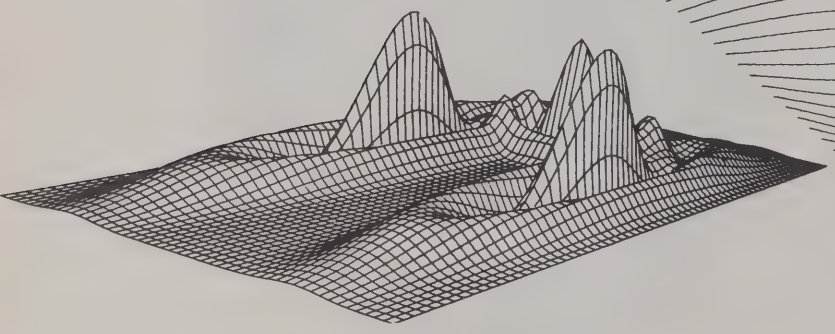
One way to create computer art is to write a weird program whose consequences you don't fully understand, tell the computer to obey it, and look at the computer's drawing. If it looks nice, keep it and call it "art"—regardless of whether the drawing was what you intended. Maybe it was the result of an error, but who cares, as long as it looks interesting? Anything interesting is art. If the drawing "has potential" but is in some way unsatisfactory, try changing a few lines...or run the program again unchanged and hope the random numbers will fall differently. The last thing to invent is the title. Whatever the drawing reminds you of becomes the title.

That method may seem a long way from DaVinci, but it's how most computer art is created. The rationale is: don't over-plan...let the computer "do its own thing"; it will give you art that escapes from the bounds of human culture and is therefore more revealing.

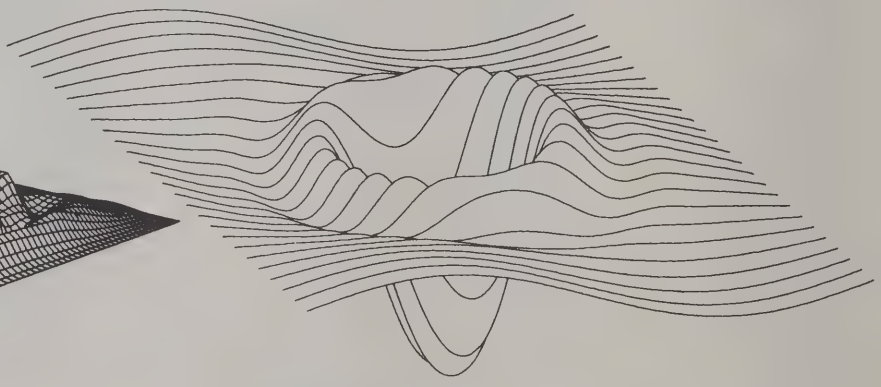
THREE-DIMENSIONAL DRAWING

The computer drew these three-dimensional surfaces:

THREE PEAKS
by John Szabo

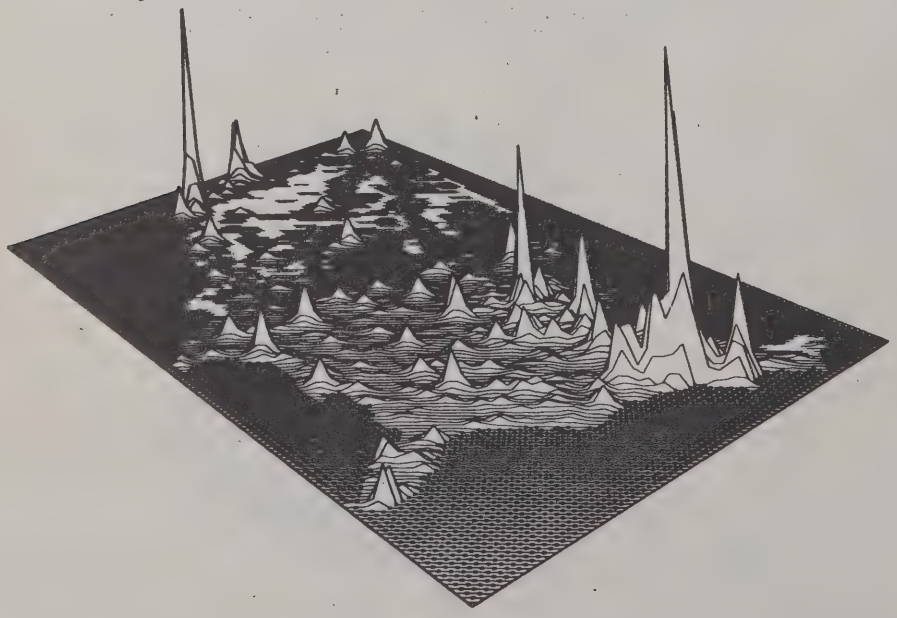


DIP

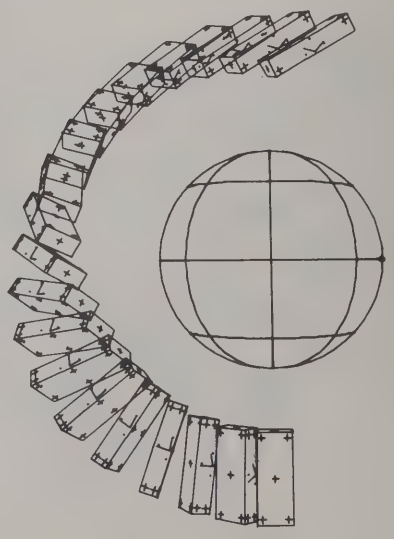


Those were done for the sake of art. These were done for the sake of science:

POPULATION DENSITY IN THE U.S.
by the Harvard University Mapping Service



ROTATION AND REVOLUTION
OF A COMMUNICATIONS SATELLITE
by Bell Telephone Laboratories



The hardest part about three-dimensional drawing is figuring out which lines the computer should *not* show, because they are hidden behind other surfaces.

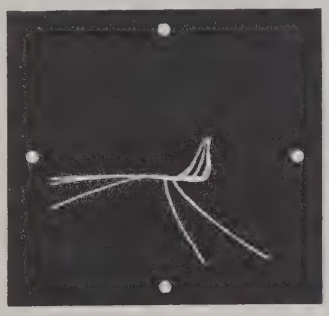
A three-dimensional drawing can be part of a much larger project—such as building a road:

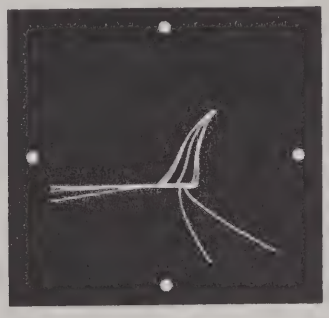
AT THE DRAWING BOARD
by Nigel Hawkes (abridged)

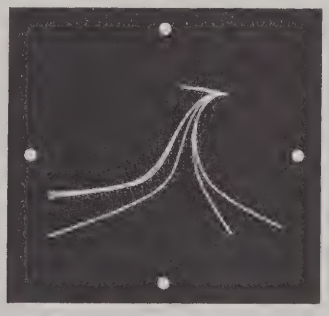
Road building is a complex operation, involving hundreds of separate problems. Since 1959 the Swedish Government and the firm Nordisk ADB of Stockholm have collaborated on a program for mechanizing these calculations. As early as 1963, more than 8000 miles of road had been planned by the system, at a savings at the planning stage of 20%.

The process starts with a series of aerial photographs of the area over which the road will run. From these, the rise and fall of the land can be analyzed stereoscopically with remarkable accuracy—said to be within four inches. The information from the pictures is fed into the computer, which calculates the amount of "cut" and "fill" needed, and how earth moved from one point can be used as fill at another. The computer allows for variations in the cost of the land, in the total length of the route, and other factors. By taking these into account, it devises the cheapest possible route between two points.

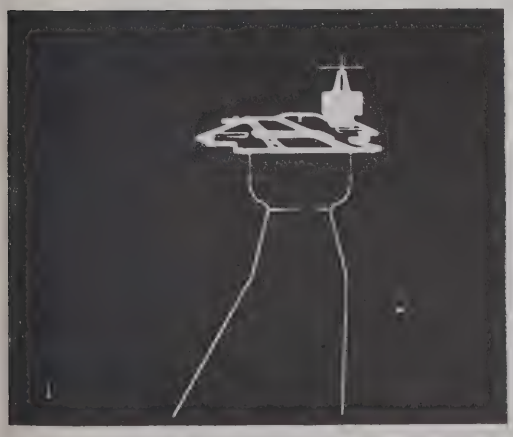
On a graphics terminal, the computer draws pictures of what the road will look like at one-meter intervals. By filming each of these pictures as a separate frame, it is possible to "drive" along the imaginary road before it is built. This is not a trivial exercise; occasionally, slopes and bends conspire to give the driver the illusion that the road is slipping sideways away from him. By observing this effect before the road is built, it can be avoided in the final design.

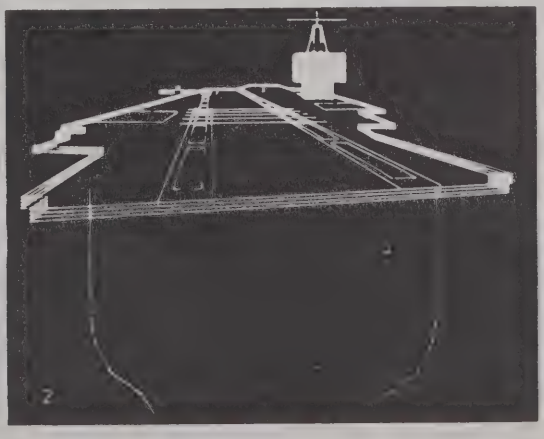


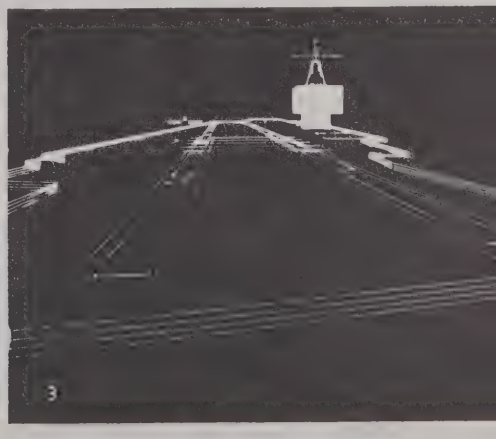


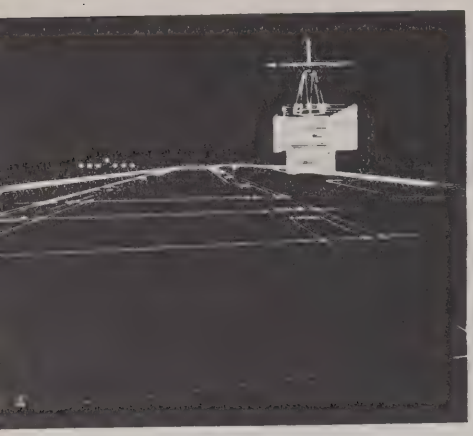


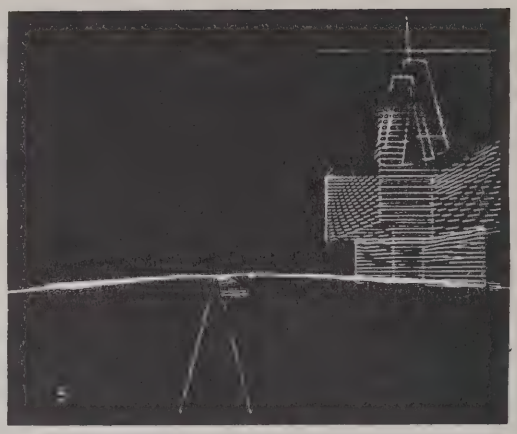
At the University of Utah, Ivan Sutherland made the computer show what happens when you try to land an airplane on an aircraft carrier. As your plane approaches the carrier, you can adjust its speed and angle by typing at the keyboard.

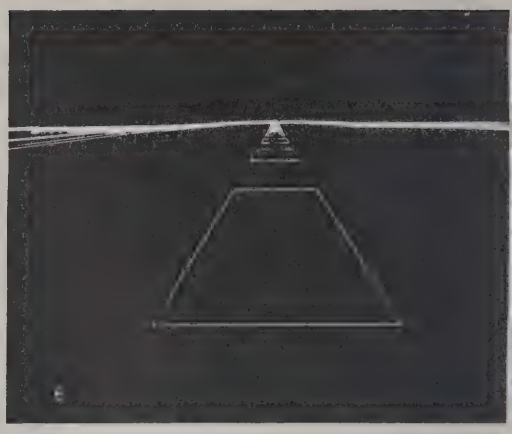












TUTORS

Judith Harris, John Swets, and Wallace Feurzeig made the computer run a logic game. The computer thinks of a letter of the alphabet; your goal is to discover which letter it is, as quickly as possible. You can ask the computer questions about the letter's shape.

In this example, the human begins by asking whether the letter has any curves:

Curves?
One.
Loose ends?
Two.
Obliques?
Zero.
C.
You don't have enough information yet to get the right answer. How do you know it isn't J, for example?
J.
You don't have enough information yet. How do you know it isn't C, for example?
Horizontals?
Zero. That was a wasted question. It did not eliminate any possibilities.
Verticals?
Zero. Good question.
G.
Wrong. Your guess is inconsistent with the information you've been given. You must have forgotten that the right letter hasn't any HORIZONTAL LINES. The letter you guessed can't be right—it has too many LOOSE ENDS.
Curves?
The answer is still: one.
X.
Wrong. Your guess is inconsistent with the information you've been given. I think you only made that guess to see what I would say.
C.
Yes, you've done it. Good work. If you want to play again, start right in.

Notice that the human can input either a guess (followed by a period) or a question (followed by a question mark). Possible guesses are *A*, *B*, *C*, *D*, etc. Possible questions are *Curves*, *Loose ends*, etc.; the human is given a list of possible questions before he runs the program.

That logic game was modified, to form a lesson about medicine. The medical case was designed by Preston Munter, of the Harvard Health Center; the programming was done by John Swets, Wallace Feurzeig, and Myra Breen. The object of the "game" is to diagnose an illness correctly, by performing sufficient tests. The human can input either a diagnosis (*Measles.*) or a request for a test (*X-ray?*). Before running the program, the human is given a list of 35 possible diagnoses and 40 possible tests.

Here are excerpts from a long RUN:

Please type your name, the date, and a slash.
Myra Breen, August 1963/
It is mid-December. Late this afternoon, a patient was admitted to the hospital. He is a well-developed, well-nourished, 34-year-old Negro coal miner. He was brought in by his wife, who reports that he has had "the grippe" for about 4 days. During the past 3 days she has given him fluids and aspirin every 6 hours with no sign of improvement. Today he suddenly became much worse. For the past 2 days he had complained of feeling very warm and of pain in his belly. This morning he had a severe shaking chill that was so bad his teeth rattled.
...
Begin the physical examination as soon as you are ready.
Physical exam?
Fine. What in particular would you like to know?
Chest exam?
It is common practice in the medical profession to begin an examination by checking the patient's general appearance.
General appearance?
Acutely prostrated, lying in bed on his right side. Looks acutely ill, sweaty, apprehensive, but alert. Paroxysmal cough, bringing up a little rusty streaked sputum. Breathing with difficulty.
EENT exam?
Check all of the vital signs first!
Temperature?
103.
Blood pressure?
150/84.
Pulse?
100 a minute. Occasional extra systole.
...
Abdominal exam?
Diffusely tender, some splinting noticed in lower right quadrant and occasionally radiating to right flank area.
The grippe.
There's insufficient information for any conclusive diagnosis at this time. You're not being careful!
Musculo-skeletal system?
Essentially negative, except for generalized weakness and aching.
Extremities?
Normal, except nail beds are questionably cyanotic but difficult to determine because of patient's skin color.
Rectal exam?
You're not being careful! The patient is too acutely ill for a rectal examination.
Laboratory reports?
Fine. Which lab tests would you like to have done?
RBC?
A red blood count will be done. Results will be sent to you as soon as possible.
...
Would you like to alert the night nurse to set up the lab tests you'll want to see tomorrow?
Yes.
Good. That will speed things up. However, there's nothing further that can be done tonight to help the patient. Let's call it quits for now. Perhaps you'll be in a better position to diagnose the case tomorrow after the first lab reports come in and you have reexamined the patient. Have a good night's sleep, and when you're back on duty tell me to "proceed."

Proceed.
It's morning and you're making your rounds. You've come to our patient and are given the night nurse's report: "Temperature remained high but patient spent a reasonably comfortable night. Cough diminished but persists. Occasional traces of blood in sputum. Patient still dyspneic and still has pain in abdomen. The second series of lab tests was initiated." Some of the first series of lab reports have just come in: urine exam, insufficient fluid for analysis; RBC, 5 million; WBC, 15500; differential, shift to left.
...
You may be prepared to make the diagnosis now. Are you?
No.
Fine. Nevertheless, we've got a fair amount of information—at least enough to raise some, perhaps several, diagnostic possibilities. Have you any in mind?
Yes.
Name one.
Pulmonary infarction.
Fine. Pleural pain, dyspnea, fever, and the physical exam signs and symptoms are certainly consistent with pulmonary infarction. Do you think that shaking chills and the presence of rusty sputum further support this diagnosis?
No.
Right. Though the white count is consistent, isn't it?
No.
Wrong. The white count is consistent. This should further support your diagnosis. We've considered one possibility. Do you have another diagnosis in mind?
No.
In that case I'd like to talk about viral pneumonia. The tachycardia, high WBC, elevated respiratory rate, shaking chills, bloody sputum, and severe pleural pain all lend weight to that diagnosis—right?
Yes.
You're way off base. There's not an iota of truth to any part of what I just said; in fact, the opposite of what I said is true. You'd better review some elementary medicine, doctor.
...
Pleural effusion.
A good contender, but let's be careful. What evidence would rule it in or out?
Positive evidence of another disease.
Is there such evidence?
Possible evidence though inconclusive.
Since you aren't sure, continue the examination until you have conclusive evidence one way or the other.
RBC?
An RBC is not indicated at this time. It's an unnecessary expense for a poor patient.
WBC?
22500.
Staph pneumonia.
Fair enough. But where is the guilty organism? When you have isolated staphylococcus aureus in significant numbers in your samples of this patient's fluids or excretions, let me know.
Pneumococcal pneumonia.
Are you absolutely sure?
Yes.
You can't be, until you have irrefutable evidence. What would help you nail down the diagnosis?
Evidence from sputum smear and culture.

Good. Anything else?
No.
I think you might also want evidence from x-ray findings. Go ahead and we'll see what happens.
X-ray?
Not ready yet. Try the blood lab.
Differential?
Marked shift to left.
...
X-ray?
The x-ray shows: area of consolidation, right lower lobe, consistent with pneumatic process. This is a good time to review the total findings and to make another tentative diagnosis—if you like.
Evidence of fluid?
No evidence of fluid.
Sputum smear?
Gram-positive rods.
Strep pneumonia.
Fine, but how can you distinguish this from any other bacterial pneumonia? Wait until you have an irrefutable case.
Viral pneumonia.
But what about shaking chills, rusty sputum, elevated WBC, etc.—all the evidence of an acute febrile bacterial process? You are in error, doctor.
Sputum culture?
Presence of pneumococci.
Strep pneumonia.
Incredible! You have disregarded the positive findings that should clearly have established the correct diagnosis.
Pneumococcal pneumonia.
Very good. It's obvious now, though there were some confusing signs and symptoms in the picture. You've seen your way through to the correct diagnosis.

Bruce Jones programmed the computer to teach you about religion. Here's an example:

This program is designed to teach you some things about the so-called "documentary hypothesis" as formulated by Julius Wellhausen and others. First, what is your name?
George
All right, George, who wrote the first five books of the Bible, according to tradition?
Moses
You are off to a good start. Do you know what those first five books are called? They are called...
the Pentateuch
Right you are. Torah is the traditional Jewish name for them. They are also called the Pentateuch ("penta" = 5). People very early noticed that there was a problem with saying that Moses wrote the Torah. If you already know the basis of their suspicion, tell me.
Some things in the text reflect later conditions, after Moses.
You are right, if course, but there was an even more obvious problem. Try again.
I don't know.
At the end of the Pentateuch the author says, "Moses...died there." Does that seem strange to you?
yes
Why?
Moses could not have written about his own death.
O.k. Also, right after the report of Moses' death and burial (Deut. 34), the text says, "No man knows the place of his burial TO THIS DAY." How do those last three words strike you?

They seem later than the event they are talking about.
A good observation, George. Many people considered these words to be anachronistic. That is, "this day" is sharply distinguished from the time of the burial. It seems likely, therefore, that the time of writing was much later than the time of burial. Again (so the argument goes), Moses was not the author. Would these observations about the end of Deuteronomy convince you that Moses did not write the Torah?
yes
Why do you say that?
A man cannot write his own obituary.
Probably not many people would have been convinced if our evidence were so limited. After all, Moses could have written everything except the last chapter. However, as early as the 18th century, people noticed another problem....

The computer searched through the human's input, to see whether he used words indicating a correct answer. The computer never said a flat "Wrong", since religious answers are a matter of personal belief, and since the human might be smarter or weirder than the computer program was prepared for.

INTERPERSONAL RELATIONSHIPS

The computer can imitate interpersonal relationships. Here's a report by Nigel Hawkes (abridged):

John Loehlin at the University of Texas devised a program that imitates human personality. In honor of the author of *Brave New World*, he calls the model Aldous. It contains five subroutines, concerned with recognition, emotional reaction, action selection, introspection, and learning, and two memories, a small immediate one and a larger long-term one. When presented with an input, Aldous responds to it "emotionally" with fear, anger, or attraction, and generates actions of withdrawal, approach, attack, conflict, or indifference. Aldous's personality can be altered by modifying the subroutines. Thus Aldous programmed one way will respond warmly to emotion ("romantic Aldous") or to abstract or concrete aspects of a situation ("intellectual Aldous") or alternatively be dominated by the past ("conservative Aldous").

Objects presented to Aldous must be described by a simple numerical code, so that Aldous does not require complex recognition equipment. When he is presented with the same object four times in a row, he gets attached to it and his attitude is positive. When presented with a new object which is very similar, his positive emotions carry over, and he approaches it with confidence; but if it does not have the same favorable consequences to him, he learns eventually to be hostile to it.

Loehlin programmed two different Aldous models into the computer at once and made them interact with one another. The social relationship turned out to be rather delicate. If either model started out in a negative frame of mind, the relationship tended to turn rapidly into mutual hostility. If both models were favorably disposed, a harmonious relationship might develop, but it was far from certain. It tended to go through phases—a warm period to start with (the honeymoon phase) followed by a phase in which each model had to cope with hostility from the other, and finally a stable relationship with increasingly positive attitudes from both models.

In another experiment, Loehlin had one Aldous interact with another programmed to produce unshakeably positive responses ("Saint Aldous"). Violent aggression resulted.

In an experiment where one model played the part of a psychotherapist, the patient, initially strongly negative, was eventually brought round to a positive attitude, at least towards the therapist. Replacing the therapist by an ordinary Aldous was not easy, however, and another period of therapy was needed before a stable and positive relationship could be established.

In a model called Homunculus, produced by John and Jeanne Gullahorn of Michigan State University, three-person interactions are possible. The model uses a theory of human relations devised by George Homans. When three unacquainted models are "introduced", they typically form a pair and an isolate; when the models have prior knowledge about one another, more complex interactions may occur.

ROBOTS

In 1962 at MIT, Heinrich Ernst connected the computer to a mechanical hand that could feel. He made the hand build objects out of blocks, and made it put blocks into boxes.

* * *

One of the most famous robots is a guy named "Shakey", built at the Stanford Research Institute (SRI) in 1970. His eye contains a television camera (optical scanner). Instead of legs, he has wheels. Instead of arms, he has antennae (for feeling) and a bumper (for pushing). His brain is a computer: instead of carrying it around with him, he leaves it in another room and communicates with it by wireless methods.

To see how he works, suppose you type this message on his computer's terminal:
Push the block off the platform.
He begins by looking for the platform. If the platform is not in the room, he goes out to the hall, and steers himself through the hall (by looking at the baseboards) until he arrives at the next room. He peers into the room, to see whether it contains a platform. If not, he hunts for another room. When he finally finds a room containing a platform with a block on it, he tries to climb onto the platform to push the block off. But before climbing the platform, he checks the platform's height. If it is too high to get onto easily, he looks for a device to help him climb it. For example, if a ramp is lying in the room, he pushes the ramp next to the platform and then wheels himself up the ramp. Finally, he pushes the block off.

He can handle unexpected situations. For example, while he's getting the ramp, suppose you pull the platform to a different place. That doesn't faze him: he hunts for the platform again, and then pushes the ramp to it.

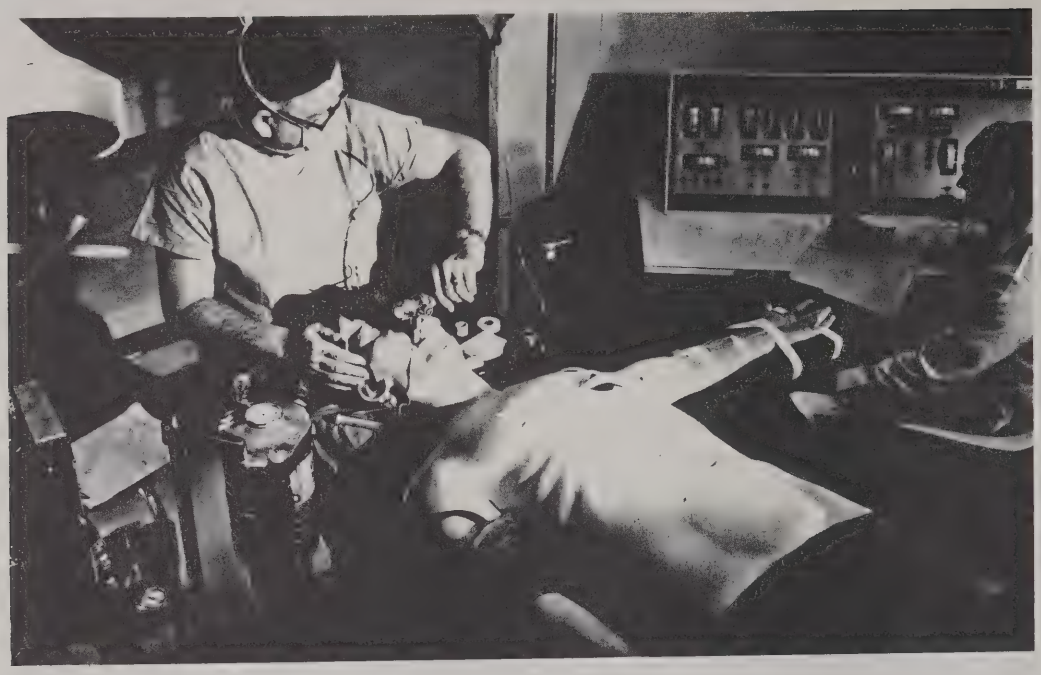
In 1971, Shakey's powers were extended, so he can handle commands such as:
Turn on the lightswitch.
If the lightswitch is too high for his bumper to reach, he looks for a device to climb onto, such as a box. If he finds a box that looks helpful, he climbs onto it, to check whether it is tall enough; if it is, he climbs off, pushes it to the lightswitch, climbs on it again, and finally flicks the switch.

Another task he can handle is:
Push three boxes together.
He finds the first box and pushes it to the second. Then he finds the third box, and pushes it to the second also.

He understands over 100 words. Whatever command you give him becomes his "goal", and he must reason out how to accomplish it. He might discover that to accomplish it, he must accomplish another goal first—for example, to move the block off the platform, he must first find the platform; to do that, he might have to look in another room; to do that, he must leave the room he's in; to do that, he must turn his wheels.

* * *

Here's a picture of a robot named Simulator One:



In the picture, a doctor is taking Simulator One's blood pressure and pulse. Another doctor is watching the computer console.

Simulator One is a model patient. He can blink, breathe, cough, vomit, respond to drugs, and even die. He is used in med school, to train doctors how to administer anaesthetics during surgery.

* * *

This report (abridged) is by Bertram Raphael, the director of the SRI Artificial Intelligence Center:

Here's what robots were capable of doing a few years ago.

At Hitachi Central Research Laboratory a TV camera was aimed at an engineering plan drawing of a structure. A second camera looked at blocks spread out on a table. The computer "understood" the drawing, reached towards the blocks with its arm, and built the structure.

At MIT, the camera was not shown a plan; instead, it was shown an example of the actual structure desired. The computer figured out how the structure could be constructed, and then built an exact copy.

At Stanford University, the hand obeyed spoken directions. For example, if someone said into the microphone, "Pick up the small block on the left," that is precisely what the arm would do.

In Scotland at the University of Edinburgh, a jumble of parts for two wooden toys was placed on a table. "Freddy," the Edinburgh robot, spread out the parts so that it could see each one clearly, and then, with the help of a vise-like work station at one corner of the table, assembled first the toy car and then the toy boat.

Recently, robot researchers have built robots that can perform truly practical tasks. For example:

At Stanford the system that used to stack toy blocks can now assemble a real water pump.

At SRI a computer-controlled arm with touch and force sensors can feel its way as it packs assembled pumps into a case.

At MIT programs are under development to enable a computer to inspect and repair circuit boards for use in computers, TV sets, and other electronic equipment.

* * *

Not all robots involve computers. Here's an example of a noncomputerized robot (reported by James Slagle, abridged):

A. George Carlton, John G. Chubbuck, and others at the Applied Physics Laboratory of Johns Hopkins University built a very interesting machine called The Beast. It is a battery-operated cylinder on wheels that is 18 in. in diameter. It has tactile, sonar, and optical apparatus. The sonar permits The Beast to find its way down the center of the hall. When its battery becomes sufficiently run-down, The Beast optically looks for an electric outlet and plugs itself in to recharge its battery. The Beast was often let loose to roam in the halls and offices at the Applied Physics Laboratory in order to see how long it could survive without "starving." Once it survived 40.6 hr. Many a new and unsuspecting secretary has been startled when The Beast entered her office, plugged itself into an electric outlet, and then departed. When it feels a step down, it knows enough to turn around, so that it does not fall downstairs. This logic however sometimes causes it to starve when it encounters a raised threshold. After getting on the threshold, it thinks it is about to fall, and so it turns around. After turning around it again thinks it is going to fall, and so it turns back and forth until it starves. It also starved when some workmen changed all the outlets from the flush to the projecting type. To cope with the new situation, the researchers changed some of the circuitry.

* * *

A recent newspaper article said that in Japan robots are being used in many practical ways. One robot welds arcs, reducing the time by 90%. Another grasps an object, determines the best way to pack it in a box, and does the packing; it uses television cameras and delicate arms. Another washes windows. Another wiggles a rod to catch a fish, takes the fish off the hook, dumps it into a bin, and returns the line to the water. Another directs traffic. Talking robots are being used instead of kimono-clad females in inns and restaurants.

Commenting on the quality of life in Japan, the article went on to say that people are buying whiffs of oxygen from vending machines.

The article was tacked on the bulletin board at the MIT Artificial Intelligence Laboratory, together with this graffito about how the Japanese robots would act differently if they were as smart as people....

HUMAN: Weld these parts.
ROBOT: The steel in those parts is grossly inferior. They must have been made in the U.S. Send them back. Also, have my welding machine tuned up.
HUMAN: Pack those widgets.
ROBOT: Can I break them in half first?
HUMAN: No.
ROBOT: "No" is not an allowed answer. I only have small shipping boxes. (Proceeds to break widgets in half and stuff them into boxes.)
HUMAN: Wash those windows.
ROBOT: What?? And get my precious electrical and mechanical components all wet??
HUMAN: Catch a fish.
ROBOT: (Proceeds to catch a fish, take it off the hook, and throw it back.) Okay.
HUMAN: What did you throw it back for?
ROBOT: It was under the size limit. Anyway, it was full of mercury.
HUMAN: Direct that traffic.
ROBOT: Will I get hazardous duty pay for this work?
HUMAN: Greet those customers.
ROBOT: Sorry, my shift is over.
HUMAN: (Ready to faint.) Give me some air.
ROBOT: Use that "machine" over there. (Points to vending machine.)

* * *

The American oil industry is already using robots to explore the ocean floor. Similar underwater robots are being requested by the Navy and the U.S. Geological Survey. Robots are also being requested for work in coal mines, outer space, and other inhuman environments.

BARTENDERS

Many bar owners don't trust the bartenders they hire. They claim the bartenders give too many free drinks to friends, steal money from the till, and put too much or too little liquor in the drinks.

To solve the problem, many bars now contain a computer that mixes and pours drinks. The computer mixes accurately. Although the computer is run by the bartender, the computer keeps an accurate record of how many drinks it makes, so there is little chance for cheating. The computer also keeps track of the inventory.

The computers are manufactured and sold by National Cash Register Co. (Dayton, Ohio), Bar Boy Inc. (San Diego, California), Electronic Dispensers International (Concord, California), and Anker-Werke (Germany). Prices range from $600 to $15000. Holiday Inns has been developing its own model.

DOCTORS

This article (abridged) appeared in *The Times*:

DIAGNOSIS BY COMPUTER MORE ACCURATE
BUT DOCTORS STILL NEEDED

A medical diagnostic system designed at Leeds University has proved more accurate than doctors in assessing the most likely cause of acute abdominal pain among patients admitted to the university's department of surgery.

Last year 304 such patients were admitted to the unit, and the computer's diagnosis proved correct in 92 percent of the cases, compared with 80 percent accuracy by the most senior doctor to see each case.

After each patient had been seen by the doctor and examined, the doctor's findings were passed on to a technician, who translated them into language used by the computer. The computer would list the likely diagnoses in order of probability. If the computer and the doctor in charge of the case disagreed, the computer would on request suggest further investigations that might be useful.

In the year-long trial the computer's diagnoses proved correct in 279 cases. In fifteen it was wrong, in eight the patient's condition was not included in the diseases considered by the computer, and in two no computer diagnosis was made because the doctors concerned with the case disagreed about the findings.

Whereas the computer advised an operation on six occasions when it would have proved unnecessary, in practice 30 such operations were carried out on the basis of the surgeon's own judgment. The computer accurately classified 84 of the 85 patients with appendicitis, compared with 75 by the doctors, and its suggestion that no operation was necessary proved correct on 136 out of 137 occasions.

The computer is reliable only if accurate data are fed into it on the basis of the doctor's interrogation and examination of the patient.

This article (abridged) is from the magazine *Computers and Automation*:

COMPUTER POISON CONTROL CENTER
OPENED BY CHILDREN'S MERCY HOSPITAL

A computerized poison control center, designed to help save children's lives by quickly identifying poisons they swallow, has been opened by The Children's Mercy Hospital, Kansas City, Mo. The computer stores information on drugs, household products, and chemicals that children may find and swallow. The computer is programmed to accept an inquiry—for example, the name of a household cleaning product—and to return to the terminal within four seconds detailed information on the poison, including symptoms and suggested treatment. The system does not replace clinical judgment, but does save valuable time in locating the requested poison.

In an experiment at another hospital, each patient was given a computer terminal and could use it to communicate with the computer about his illness. Afterwards, most of the patients said they'd rather deal with the computer than with human doctors.

This report is by James Martin and Adrian Norman:

> The Los Angeles County General Hospital Shock Research Unit is concerned with circulatory shock, which results in low blood pressure, low blood flow, and an extremely unstable circulatory system. The proportion of circulatory-shock patients who die is large. A computer was installed to improve the monitoring and care of these patients. A patient in this ward has many instruments attached to him: six thermistors measure the temperature at various parts of his body; pressures are measured in the circulatory system; catheters are inserted into an artery and a vein; and a urinometer measures liquid waste. The instruments are all directly connected to the computer, and other details about the patient are given to the machine by the nurse. The computer prints out a log about the patient's condition, and if any measurement falls outside critical limits set by the doctor for that patient, the computer notifies the ward staff.

JOBS

There are several kinds of computer-related jobs. I'll list them in order, from the lowest paid to the highest.

THE KEYPUNCHER is a typist; she types cards to feed into the computer. To get a decent salary, she must type quickly and accurately. She does not have to be creative: other people will tell her exactly what to type. I say "she" because most keypunchers are poorly paid women. They are often called *the keypunch girls*. Since keypunch machines are being replaced by terminals, and the word *girl* is considered sexist, the term *keypunch girls* is changing to *data entry personnel*. The job is secure, but is repetitive, mind-deadening, and offers little chance for advancement. When the keypuncher has finished typing her deck of cards, she hands it to *the operator*.

THE OPERATOR is a baby-sitter; his job is to take care of the computer. When it is hungry, he feeds it cards. When it spits out answers, he hands the answers to the people who requested them. If the computer starts having a fit, he makes a quick attempt to calm the computer down; if he is unsuccessful, he calls *the technician*.

THE TECHNICIAN is a repairman; he fixes the computer when it is broken. Like any good doctor, he gives the computer a periodic "check-up", to make sure it is working. Usually the check-up is given once a week, in the morning. It is called *preventive maintenance* and consists of *diagnostic tests*. The technician must know what the *operator* knows, and in addition must understand *digital electronics*, especially circuit diagrams, so he can take the computer apart. If the computer is to be attached to another machine, the technician will do the wiring. The computer center usually needs the technician only once or twice a week; the other days of the week he visits other computer centers. His salary is paid by the company that manufactured the computer. He is sometimes called a *field service engineer*.

THE PROGRAMMER is a translator; he translates English instructions into language the computer can understand. In Part 1 of *The Secret Guide to Computers*, I explained how to be a programmer; part 3 will explain further. *Commercial programmers* write programs for businessmen; *scientific programmers* write programs for scientists. Commercial programmers and scientific programmers are both called *applications programmers*. The opposite of an applications programmer is a *systems programmer*, who writes programs that help programmers write programs.

The typical programmer works in a company for 2 years and then switches to a different company, to gain new experience and a jump in salary. By so doing, the typical programmer increases his salary at an average rate of $2000/year. ($1300 of that is because of his increased expertise, and the other $700 is because of inflation.)

In 1975, 81 programmers at 14 computer centers were interviewed. They survey concluded that the average programmer has only 2 years of experience, only 2 years of college education, has programmed only 2 applications, knows only 2 computer "languages", is introverted, sloppy, and inflexible, feels he is "in over his head", and is undermanaged.

Another survey shows that only 10% of programmer time is spent in writing (*coding*) programs. 20% of programmer time is spent in *planning* what to write (that's called *program design*); another 20% is spent in *testing* the program to see whether it actually works, and removing the errors; the remaining 50% is spent in changing older programs (that's called *program maintenance*). So it's not surprising that the average programmer writes less than 10 new lines per day. In some companies, the figure is as low as 6 lines per day.

One reason for the low level of production is that programmers tend to work on *projects* in *teams*; each programmer spends much time communicating with other members of the team, to make sure his program will work well when combined with theirs.

There are several "steps up" a programmer can take. A *programmer/analyst* has the additional responsibility of figuring out *what* to program. A *project leader* or *team leader* is in charge of a team of programmers.

THE DATA PROCESSING MANAGER, also called the COMPUTER CENTER DIRECTOR, is the boss. His job is to make sure everyone in the computer center is working well and is happy. Once a month, he writes a newsletter, explaining what's new in the computer center. He determines what additional computer equipment is needed, and asks the president for the funds. In a large computer center, he is assisted by several other managers (an *operations manager*, a *systems manager*, a *manager of user services*, etc.).

JARGON

The study of computers is called *computer science*. Using a computer, especially to solve problems in business (such as payroll and inventory), is called *data processing* (abbreviated *DP*) or *automatic data processing (ADP)* or *electronic data processing (EDP)* or *information processing*. *Semantic information processing* means using a computer to understand ordinary English (see pages 2-11).

A distinction is often made between *business applications* and *scientific applications*. Business applications usually involve lengthy input and lengthy output, but the computation is simple-minded (no square roots or trigonometry). Scientific applications involve advanced math, but the input and output are usually brief. In between lie *statistical applications*, which involve moderately complicated math (square roots but not trigonometry) and moderately lengthy input and output.

Using the computer to control another machine is called *numerical control (NC)*. Now computers are controlling assembly lines, traffic lights, pinball machines, microwave ovens, and electronic organs. Instead of putting the computer next to the machine it controls, the modern tendency is to put the computer *inside* the machine. That's possible, because computers have been miniaturized. A miniature computer is called a *minicomputer*; an even smaller one is called a *microcomputer*. Microcomputers can be bought for as little as $15, so it is becoming economically practical to put them inside ordinary household appliances.

A *computer manufacturer* is a company that makes computers. An *end user* is someone who buys one computer. An *original equipment manufacturer (OEM)* is a company that buys 100 or more computers, puts each computer into another machine, and then sells the computerized machines. *OEM equipment* means computers sold to OEMs; they are sold without fancy casing, since the OEM will build his own, larger case and put additional machinery inside it.

A *Hospital Information System (HIS)* means a computerized system for handling hospital records. A *Management Information System (MIS)* means a computerized system for managing a company, with particular emphasis on long-range planning and analysis.

Information retrieval (IR) means using the computer to locate information. For example, if you go to a library to find a book, and the librarian doesn't have it, he uses a computerized IR system to find out which other library to go to. If a policeman sees a speeding car, he radios the license plate number to an IR system, which tells him whether the driver is a known criminal. If you want to reserve a seat on an airplane, the airline agent types your request on a computer terminal; the IR system tells him whether the plane is full.

The data stored in an IR system used to be called a *data bank*, but the term is changing to *data base*. The question of how to organize the data, revise it, and retrieve it is called *data base management*. There are many possible ways to organize data: you can form lists, tables, or fancier *structures*. Many colleges offer a course in *data structures*, and how to use them in *sorting* and *searching for* information.

The question of how to make the computer do advanced math (such as calculus) as accurately and quickly as possible is called *numerical analysis*; solutions to the question are called *numerical methods*.

Simulate means "imitate"; *simulation* means "imitation". I already showed how the computer simulates (imitates) a psychotherapist (p. 7), a paranoid schizophrenic in a mental institution (p. 11), driving on a road (p. 33), landing on an airplane carrier (pp. 33-34), interpersonal relationships (p. 38), and the functionings of the human body (p. 40). When the computer simulates (imitates) something, the computer's program is called the *model*. It is a *good model* if it is a good imitation.

At Dartmouth College, the computer simulates birth control. The computer asks you how old you are, and which birth control method you wish to use this year. There are nine choices (for example: pill, IUD, diaphragm, rhythm method, and "Providence"). After you type your choice, the computer computes the probabilities and may print (if you are unlucky) ***BOY*** or ***GIRL***. The computation is based, as in nature, on a combination of science and chance (random numbers). Then it asks for your strategy for the next year. The program continues until it finally prints ***MENOPAUSE***. Using the program, you can explore how different strategies will result in different numbers of children. It's safer to experiment with the program than to experiment on your own body.

That's the major use of simulation: exploring what will happen if you take certain actions. Simulation has been used to explore how traffic jams would change, if new roads were built or old roads were made one-way; how much the noise level in Boston would increase, if the noisy Concorde airplane were allowed to fly into Boston's airport; how the heroin traffic in the New York area would change, if the Bronx section were to clamp down on pushers; whether urban renewal would make New York worse, by encouraging unemployable people to move there.

The use of computers in schools is usually called *computer-assisted instruction (CAI)*. It tends to denote drill-like tutoring by the computer.

Computer-aided learning (CAL) denotes a movement away from drill, and often involves simulation. A family-planning student experiments with the program that simulates birth control. An economics student gives the computer his own plan for fixing the national economy, and watches the computer tell him how the economy only gets worse. A physics student tries to make a computer-imitation rocket land on a computer-imitation moon.

Computer-aided education (CAE) and *computer-managed instruction (CMI)* emphasize school administration, such as computing grade-point averages and giving students their schedules.

ARTIFICIAL INTELLIGENCE

Living organisms, such as man, have what is called *natural intelligence*. In many ways, computers seem to be smart too: their intelligence is called *artificial intelligence* (abbreviated *AI*). Can a computer become as smart as a person? Can artificial intelligence become as powerful as natural intelligence? The question as a long, colorful history. Here it is, in chronological order....

1. HOPE

The hope of making an inanimate object as smart as a person can be traced back to the ancient Greeks. According to Greek mythology, Pygmalion sculpted a statue of a woman, fell in love with it, and prayed to the gods to make it come to life. His wish was granted—she came to life. And they lived happily ever after.

In 1272 A.D. on the Spanish island of Majorca, Ramon Lull invented the idea of a machine that would produce *all* knowledge, by putting together words at random. He even tried to build it.

Needless to say, he was a bit of a nut. Here's a description of his personality (written by Jerry Rosenberg, abridged):

> Ramon Lull married young and fathered two children—which didn't stop him from his courtier's adventures. He had an especially strong passion for married women. One day as he was riding his horse down the center of town, he saw a familiar woman entering church for a High Mass. Undisturbed by this circumstance, he galloped his horse into the cathedral and was quickly thrown out by the congregants. The lady was so disturbed by this scene that she prepared a plan to end Lull's pursuit once and for all. She invited him to her boudoir, displayed the bosom that he had been praising in poems written for her, and showed him a cancerous breast. "See, Ramon," she said, "the foulness of this body that has won thy affection! How much better hadst thou done to have set thy love on Jesus Christ, of Whom thou mayest have a prize that is eternal!"
>
> In shame Lull withdrew from court life. On four different occasions a vision of Christ hanging on the Cross came to him, and in penitence Lull became a dedicated Christian. His conversion was followed by a pathetic impulse to try to convert the entire Moslem world to Christianity. This obsession dominated the remainder of his life. His *Book of Contemplation* was divided into five books in honor of the five wounds of Christ. It contained forty subdivisions—for the forty days that Christ spent in the wilderness; 366 chapters—one to be read each day and the last chapter to be read only in a leap year. Each of the chapters had ten paragraphs to commemorate the ten commandments; each paragraph had three parts to signify the trinity—for a total of thirty parts a chapter, signifying the thirty pieces of silver.
>
> In the final chapter of his book he tried to prove to infidels that Christianity was the only true faith.

Several centuries later—in 1726—Lull's machine was pooh-poohed by Jonathan Swift, in *Gulliver's Travels*. Gulliver meets a professor who has built such a machine. The professor claims his machine lets "the most ignorant person...write books in philosophy, poetry, politics, law, mathematics, and theology without the least assistance from genius and study." The machine is huge—20 feet on each side—and contains all the words of the language, in all their declensions, written on scraps of paper that are glued onto bits of wood connected by wires. Each of the professor's 40 students operates one of the machine's 40 cranks. At a given signal, every student turns his crank a random distance, to push the words into new positions. Gulliver reports: "He then commanded six and thirty of the lads to read the several lines softly as they appeared upon the frame; and where they found three or four words together that might make part of a sentence, they dictated to the four remaining boys who were scribes....Six hours a day the young students were employed in this labor, and the professor showed me several volumes in large folio already collected, of broken sentences, which he intended to piece together, and out of those rich materials to give the world a complete body of all arts and sciences...."

The word *robot* was invented in 1920 by Karel Capek, a Czech playwright. His play "R.U.R." shows a factory where the workers look human but are really machines. The workers are dubbed *robots*, because the Czech word for *slave* is *robotnik*.

His play is pessimistic. The invention of robots causes unemployment. Men lose all ambition—even the ambition to raise children. The robots are used in war, go mad, revolt against mankind and destroy it. In the end only two robots are left. It's up to them to repopulate the world.

Other sci-fi writers copied Capek's idea of robots, with even more pessimism. An exception was Isaac Asimov, who depicted robots as being loving. He coined the word *robotics*, which means the study of robots, and in 1942 developed what he calls the "Three Laws of Robotics". Here's the version he published in 1950:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence, as long as such protection does not conflict with either the First or the Second Law.

The word *cybernetics* was invented in 1947 by Norbert Wiener, an MIT professor. He defined it to be "the science of control and communication in the animal and the machine." Wiener and his disciples, who called themselves *cybernéticists*, wondered whether it would be possible to make an electrical imitation of the human nervous system. It would be a "thinking machine". They created the concept of *feedback*: animals and machines both need to perceive the consequences of their actions, to learn how to improve themselves. For example, a machine that is producing parts in a factory should examine the parts it has produced, the heat it has generated, and other factors, to adjust itself accordingly.

Wiener, like Ramon Lull, was somewhat strange. He graduated from Tufts College when he was 14 years old, got his doctorate from Harvard when he was 18, and became the typical "absent-minded professor". Many anecdotes are told about him.

For example, once he went to a conference and parked his car in the parking lot. When the conference was over, he went to the lot, but forgot where he parked his car, and even forgot what it looked like. So he waited until all the other cars were driven away, and took the car that was left.

When he and his family moved to a new house a few blocks away, his wife gave him written directions on how to reach it, because she knew he was absent-minded. But sure enough, when he was leaving his office at the end of the day, he couldn't remember where he put her note, and of course he couldn't remember where the new house was. So he drove to his old neighborhood instead. He saw a young child and asked her, "Little girl, can you tell me where the Wieners have moved to?" "Yes, Daddy," came the reply, "Mommy said you'd probably be here, so she sent me over to show you the way home."

One day he was sitting in the campus lounge, intensely studying a paper on the table. Every now and then, he would get up, pace a bit, and then return to the paper. Everyone was impressed by the enormous mental effort reflected on his face. Once again he rose from his paper, took a few rapid steps around the room, and collided with a student. The student said, "Good afternoon, Professor Wiener." Wiener stopped, stared, clapped a hand to his forehead, said "Wiener—that's the word," and ran back to the table to fill the word "wiener" in the crossword puzzle he was working.

Once he drove 150 miles to a math conference at Yale University; but when the conference was over, he forgot he had come by car, so he returned by bus. The next morning, he went out to his garage to get his car, discovered it was missing, and complained to the police that while he was away someone had stolen it.

Those anecdotes were collected by Howard Eves, a math historian.

Can a computer "think"? In 1950, Alan Turing proposed the following test. In one room, put a human and a computer. In another room, put another human, called the Interrogator, and give him two terminals—one for communication with the computer, and the other for communication with the other human—but don't tell him which terminal is which. If he can't tell the difference, the computer is doing a good job of imitating the human, and, according to Turing, we should say that the computer is capable of "thinking".

It's called the Imitation Game. The Interrogator asks questions. The human witness answers honestly. The computer pretends it is human.

To win, the computer must be able to imitate human weaknesses as well as strengths. For example, when asked to add two numbers, it should pause before answering, as a human would. When asked to write a sonnet, a good imitation-human answer would be, "Count me out on this one. I never could write poetry." When asked "Are you human," the computer should say "yes".

Such responses wouldn't be hard to program. But a clever Interrogator could give the computer a rough time, by requiring it to analyze its own thinking:

INTERROGATOR: In the first line of your sonnet which reads "Shall I compare thee to a summer's day," would not "a spring day" do as well or better?
WITNESS: It wouldn't scan.
INTERROGATOR: How about "a winter's day". That would can all right.
WITNESS: Yes, but nobody wants to be compared to a winter's day.
INTERROGATOR: Would you say Mr. Pickwick reminded you of Christmas?
WITNESS: In a way.
INTERROGATOR: Yet Christmas is a winter's day, and I do not think Mr. Pickwick would mind the comparison.
WITNESS: I don't think you're serious. By a winter's day one means a typical winter's day, rather than a special one like Christmas.

If the computer could answer questions that well, the Interrogator would have a hard time telling it wasn't human.

Donald Fink has suggested that the Interrogator say, "Suggest an unsolved problem and some methods for working toward its solution," and "What methods would most likely prove fruitful in solving the following problem...."

Turing believed computers would someday be able to win the game and therefore be considered to "think". In his article, he listed possible objections to his belief, and rebutted them....

OBJECTION: Thinking is a functions of man's immortal soul. Since computers don't have souls, they can't think.
REBUTTAL: Since God is omnipotent, He can give computers souls if He wishes. Just as we create children to house His souls, so should we serve Him by creating computers.
OBJECTION: If machines could equal us in thinking, that would be dreadful!
REBUTTAL: Too bad!
OBJECTION: Logicians have proved it's impossible to build a computer that can answer every question.
REBUTTAL: Is it possible to find a human that can answer every question? Computers are no dumber than us. And although no one computer can answer every question, why not build a succession of computers, each one more powerful than the next, so every question could be answered by at least one of them?
OBJECTION: Although computers can produce, they can't be conscious of what they've produced. They can't feel pleasure at their successes, misery at their mistakes, and depression when they don't get what they want.
REBUTTAL: The only way to be absolutely sure whether a computer has feelings is to become one. A more practical experiment would be to build a computer that explains step-by-step its reasoning, its motivations, and the obstacles it is trying to overcome, and also analyzes emotional passages such as poetry. Such a computer is clearly not just parroting.
OBJECTION: A computer can't be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make someone fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behavior as a man, or do something really new.
REBUTTAL: Why not? Although no such computer has been built yet, it might be possible in the future.

OBJECTION: The computer never does anything original or surprising. It does only what it is told.
REBUTTAL: How do you know "original" human work isn't just the growth of a seed implanted by teaching, or the effect of well-known general principles? And who says computers aren't surprising? The computer's correct answers are often surprisingly different from my rough guesses.
OBJECTION: Nerve cells are sensitive to gradual increases in electrical activity—you can feel a "little tingle" or a "mind pain" or an "ouch"—whereas a computer's logic is only binary—either "yes" or "no".
REBUTTAL: By using random numbers and other techniques, you can make the computer imitate the flexible, probabilistic behavior of the nervous system well enough so that the Interrogator can't tell the difference.
OBJECTION: Life cannot be reduced to rules. For example, if you have a traffic-light rule that says "stop when the light is red, and go when the light is green", what do you do when the light is broken, and both the red and the green appear simultaneously? Maybe you should have an extra rule saying in that case to stop. But some further difficulty may arise with that rule, and it would be necessary to create another. And so on. You can't invent enough rules to handle all cases. Since computers must be fed rules, they cannot handle all of life.
REBUTTAL: Although life is more than a simple set of rules, it might be the consequences of simple psychological laws of behavior, which the computer could be taught.
OBJECTION: Humans have extrasensory perception (ESP), and computers do not.
REBUTTAL: Maybe the computer's random-number generator could be hooked up to be affected by ESP. Or to prevent ESP from affecting the Imitation Game, put both the human witness and the computer in a telepathy-proof room.

To turn the computer into an intelligent creature, Turing suggested two different ways to begin. One would be to teach it abstract skills, such as chess. The other would be to give it eyes, ears, and other sense organs, teach it how to speak English, and then educate it the same way you'd educate a somewhat handicapped child.

Four years later—on June 8, 1954—Turing was found dead in bed. According to the police, he died from potassium cyanide, self-administered. He had been plating spoons with potassium cyanide in electrolysis experiments. His mother refuses to believe it was suicide, and hopes it was just an accident.

In 1950—the same year as Turing's article—Claude Shannon proposed a way to make the computer win at checkers, chess, and other complicated games.

To understand his method, let's try to make the computer win a game of checkers. As in all checker tournaments, one player is called BLACK, and the other is called WHITE (even though his pieces are actually red). BLACK makes the first move. When a player can jump, he must. The game ends when one of the players is unable to move (either because he has no pieces or because his pieces are blocked).

To simplify the game, we'll play on a 4-by-4 board, instead of the traditional 8-by-8. Each player has two pieces instead of twelve.

Examine the diagram on the opposite page, which shows 63 possible positions. Position #1 is the initial position, from which black will move. The three arrows coming from position #1 represent the three legal moves he can choose from. Depending on which move he chooses, the board will wind up in position #2 or #3 or #4. Which move is the best?

To find out, let's examine the consequences of each move:

If he moves to position #2, white will reply by moving to position #5 or #6 or #7.
If he moves to position #3, white will reply by moving to position #8 or #9 or #10.
If he moves to position #4, white will reply by moving to position #11 or #12 or #13.

The diagram shows all the possible ways the first five moves of the game could go. Throughout the diagram, w means white man, b means black man, ẅ means white king, and b̈ means black king. The diagram is called a *tree*. (If you turn it upside down, it looks like the kind of tree that grows in the ground.) The arrows are called the tree's *branches*. The tree's *depth* is 5.

In deciding whether black should choose position #2 or #3 or #4, the accuracy of our analysis will depend on how deep we make the tree. In this particular game, a depth of 5 is satisfactory; but in 8-by-8 checkers or chess we might have to dig deeper. In theory, we should keep digging until we reach the end of the game; but such a tree might be too large to fit in the computer's memory.

For chess, Shannon estimated that a complete tree requires 10^{120} branches. Einstein estimated that the number of electrons in the universe is only 10^{110}. If Shannon and Einstein are both right, the tree can't fit in the universe!

Having constructed our tree of depth 5, we look at the bottom positions (#42 through #63) and evaluate them, to see which ones look favorable for black. There are many possible factors to take into account: which player has control of the center of the board? which player can move the most without being jumped? and so on. But to keep matters simple, we'll consider just one factor: which player has the most men? We'll consider a king to be worth 1½ men. We'll subtract the number of white men from the number of black men.

The result of the evaluation is a number, which is called the position's *value*. If it's negative, black is losing; if it's positive, black is winning; if it's zero, the game is heading for a draw.

For example, consider position #42. Since black has one man and white has two, the value is 1 minus 2, which is -1. That's why I've written "v=-1" underneath that position. The value of each position at depth=5 is computed by that method.

For the positions at depth=4, a different method is used. For example, here's how to find the value of position #29. That position has two possible outcomes: #46 and #47. Which outcome is more likely? Since the move will be made by black, and black's goal is to make the value large, he will prefer to move to #46 instead of #47. Since the most likely outcome has a value of ½, we assign position #29 a value of ½ also.

Here's the rule: to compute the value of a position at depth=4, find the *maximum* value of the positions it points to. (The value of position #29 is the maximum value of positions #46 and #47, which is ½.)

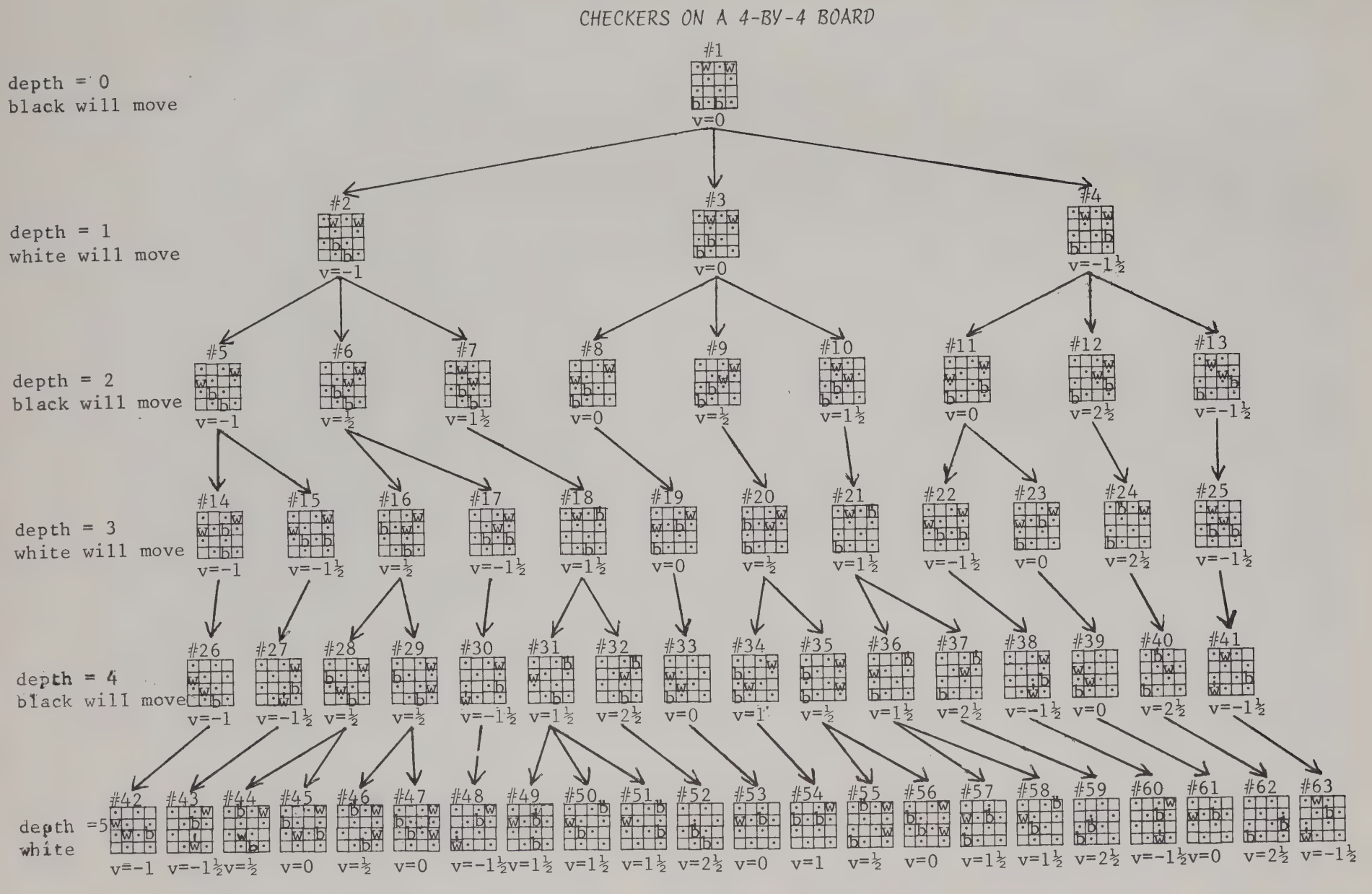
To compute the value of a position at depth=3, find the *minimum* value of the positions it points to (since it's white's turn to move, and white wants to minimize). For example, the value of position #18 is the minimum value of positions #31 and #32, which is 1½.

The values for depth 2 are computed by maximizing, and the values for depth 1 are computed by minimizing. We finally get these results:

the value of position #2 is -1
the value of position #3 is 0
the value of position #4 is -1½

Since black's goal is to maximize values, he should move to position #3. If white is also a good player, the game will probably gravitate toward position #53, a draw. If white is a poorer player, black will win.

That method of choosing the best move was proposed by Shannon, makes heavy use of minimums and maximums, and is called the *minimax* method.


CHECKERS ON A 4-BY-4 BOARD
depth = 0
black will move
depth = 1
white will move
depth = 2
black will move
depth = 3
white will move
depth = 4
black will move
depth =5
white
#1 v=0
#2 v=-1
#3 v=0
#4 v=-1½
#5 v=-1
#6 v=½
#7 v=1½
#8 v=0
#9 v=½
#10 v=1½
#11 v=0
#12 v=2½
#13 v=-1½
#14 v=-1
#15 v=-1½
#16 v=½
#17 v=-1½
#18 v=1½
#19 v=0
#20 v=½
#21 v=1½
#22 v=-1½
#23 v=0
#24 v=2½
#25 v=-1½
#26 v=-1
#27 v=-1½
#28 v=½
#29 v=½
#30 v=-1½
#31 v=1½
#32 v=2½
#33 v=0
#34 v=1
#35 v=½
#36 v=1½
#37 v=2½
#38 v=-1½
#39 v=0
#40 v=2½
#41 v=-1½
#42 v=-1
#43 v=-1½
#44 v=½
#45 v=0
#46 v=½
#47 v=0
#48 v=-1½
#49 v=1½
#50 v=1½
#51 v=1½
#52 v=2½
#53 v=0
#54 v=1
#55 v=½
#56 v=0
#57 v=1½
#58 v=1½
#59 v=2½
#60 v=-1½
#61 v=0
#62 v=2½
#63 v=-1½

2. RESULTS

After Turing and Shannon, the next person to become famous was Arthur Samuel. He spent a good part of his life (twenty years, from 1947 to 1967) trying to make the computer win checkers. He used Shannon's minimax idea, but made many improvements.

His first spectacular success came in 1962, when his program won a game against Robert Nealey, a former Connecticut checkers champion. After the game, Nealey said, "The computer had to make several star moves in order to get the win....In the matter of the end game, I have not had such competition from any human being since 1954, when I lost my last game."

Later, the computer played six more games against Nealey. Nealey won one of them; the other five were draws.

In 1965 the computer played four games against W.F. Hellman, the World Champion. The games were played by mail. Under those conditions, Hellman won all four. But in a hastily played game where Hellman sat across the board from the computer, the result was a draw.

In 1967 the computer was beaten by the Pacific Coast Champion, K.D. Hanson, twice.

In short, the computer wins against most humans and draws against most experts, though it loses to the top champions. To bring the computer to that level of intelligence, Samuel made three modifications to Shannon's method:

1. When choosing among several moves, the computer analyzes the most promising ones more deeply.
2. After computing the value of a position (by examining the positions under it), the computer writes the value on a piece of tape. If the position recurs in another game, the computer looks at the tape instead of repeating the analysis.
3. To compute the value of a position, the computer examines many factors in addition to the number of pieces each player has. The computer combines the factors, to form combination-factors, and then combines the combination-factors to form a single value. The relative importance given to each factor is determined by "experience". Samuel experimented with two forms of experience: he had the computer play against itself, and also had it analyze 250,000 moves that occurred in checker championships.

For details, see the *Commentary*.

Other programmers tried to write a similar program for chess. They had a hard time. In 1960 the best chess program that had been written was beaten by a ten-year-old novice. The first decent chess program was written in 1967 by Richard Greenblatt and his friends at MIT: it actually won a game in a chess tournament. But in most tournaments it lost. In 1970 and 1971, it lost every game in every tournament in entered. Better chess programs have been written since then, but the progress has been small. Around 1970, when the best chess program in the world played against Northwestern University students, it placed 13th out of 24.

While Samuel, Greenblatt, and others were working on checkers and chess, other branches of knowledge were explored as well. In 1956 at the University of Illinois, Lejaren Hiller and Leonard Isaacson made the computer compose its own music. There are no sharps or flats. Each not is picked by using random numbers; but before printing the note, the computer checks whether it would contradict any of the fourteen rules of counterpoint; if so, the computer picks a different note instead. Hiller says the music is "of fair quality, strongly reminiscent, if one ignores a certain monotony in rhythm, of passages from Palestrina."

In 1957 they tried another experiment: they allowed sharps and flats, and fed in only four of the fourteen rules. The result was chromatic music. According to Hiller, it "recalled passages from, say, a Bartok string quartet." They also made the computer compose twelve-tone music in the style of Schoenberg.

Most research in artificial intelligence was done at the Massachusetts Institute of Technology (MIT) and the Carnegie Institute of Technology (CIT, now called Carnegie-Mellon University). At Carnegie, the big names were Allen Newell and Herbert Simon. They invented the *protocol method*. In the protocol method, a human subject is told to solve a tough problem and, while he is solving it, to say at each moment what he is thinking. A transcript of his train of thought is recorded and called the *protocol*. Then an attempt is made to program the computer to imitate that train of thought.

Using the protocol method, Newell and Simon produced programs that could "think like humans". The thinking, like human thinking, was imperfect. The purpose of their research was not to make the computer a perfect thinker, but rather to gain insight into human thinking. Their point of view was: if you think you really understand human psychology, go try to program it. Their attempt to reduce human psychology to computer programs is called *mentalism*, and has replaced Skinner's stimulus-response behaviorism as the dominant force in psychology today.

There have been several attempts to make the computer do abstract mathematics. In 1957 Newell, Simon, and Shaw used the protocol method to make the computer prove theorems about symbolic logic, such as "Not (p or q) implies not p". In 1959 and 1960, Herbert Gelernter and his friends made the computer prove theorems about Euclidean geometry, such as "If the segment joining the midpoints of the diagonals of a trapezoid is extended to intersect a side of the trapezoid, it bisects that side." In 1961, MIT awarded a Ph.D. to James Slagle for making the computer compute indefinite integrals, such as $\int \frac{x^4}{(1-x^2)^{5/2}} dx$; the computer gets the answer, which is

$$\arcsin x + \frac{\tan^3 \arcsin x}{3} - \tan \arcsin x + c.$$

Each of those programs works by drawing a tree inside the computer's memory. Each branch of the the tree represents a possible line of attack. The computer considers each branch and chooses the one that looks most promising.

A better symbolic-logic program was written by Hao Wang in 1960. Wang's program doesn't need trees; it always picks the right attack immediately. It is guaranteed to prove any theorem it is handed, whereas the program by Newell, Simon, and Shaw got stuck on some hard ones.

A better indefinite integration program was written by Joel Moses in 1967 and further improved in 1969. It uses trees very rarely, and solves almost any integration problem.

A program that usually finds the right answer but might fail on hard problems is called *heuristic*. A heuristic program usually involves trees. The checkers, chess, and geometry programs are heuristic. A program that is guaranteed to always give the correct answer is called *algorithmic*. The original symbolic-logic program was heuristic, but Wang's improvement is algorithmic; Moses's indefinite integration program is almost algorithmic.

In 1957 Newell, Simon, and Shaw began writing a single program to solve *all* problems. They called the program GPS (General Problem Solver). If you feed the program a goal, a list of operators, and associated information, the program will tell you how to achieve the goal by using the operators.

For example, suppose you want the computer to solve this simple problem: a monkey would like to eat some bananas that are too high for him to reach, but there is a box nearby he can stand on. How can he get the bananas?

Feed the GPS program this information....

NOW: monkey's place = $place_1$; box's place = $place_2$; contents of monkey's hand = empty
WANT: contents of monkey's hand = the bananas
DIFFICULTIES: contents of monkey's hand is harder to change than box's place, which is harder to change than monkey's place

ALLOWABLE OPERATOR	DEFINITION
climb box	before: monkey's place = box's place after: monkey's place = on the box
walk to x	after: monkey's place = x
move box to x	before: monkey's place = box's place after: monkey's place = x; box's place = x
get bananas	before: box's place = under the bananas; monkey's place = on the box after: contents of monkey's hand = the bananas

GPS will print the solution:

walk to $place_2$
move box to under the bananas
climb box
get bananas

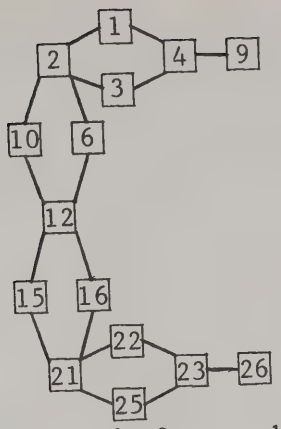
The GPS approach to solving problems is called *means-end analysis*: you tell the program the means (operators) and the end (goal). The program has proved theorems in symbolic logic, computed indefinite integrals, and solved many famous puzzles, such as "The Missionaries and the Cannibals", "The Tower of Hanoi", and "The 5-Gallon Jug and the 8-Gallon Jug". But the program works slowly, and you must feed it a lot of information about the problem. The project was abandoned in 1967.

Also included under the heading *artificial intelligence* is the attempt to make the computer understand English, Russian, and other languages used by humans. Such languages are called *natural languages*, as opposed to the restrictive *computer languages* in which programs are written. I discussed natural languages on pages 1-11.

Another large topic in artificial intelligence is *computer vision*: making the computer see. The first problem tackled was *pattern recognition*: making the computer read handwritten printed letters. The problem is hard, because some people make their letters very tall or wide or slanted or curled or close together, and the pen may skip. Reasonably successful programs were written, although computers still can't tackle script. Interest later shifted to *picture processing*: given a photograph of an object, make the computer tell what the object is. The problem is hard, because the photo may be taken from an unusual angle and blurred, and because the computer gets confused by shadows. *Scene analysis* is even harder: given a picture of a group of objects, make the computer tell which object is which. The problem is hard, because some of the objects may be partly hidden behind others, and because a line can have two different interpretations: it can be a crease in one object, or a dividing-line between two objects. Most of the research in picture processing and scene analysis was done rather recently, from 1968 to 1972.

Now there are four major places where artificial-intelligence research is done: MIT, Carnegie, Stanford, and Edinburgh (Scotland).

Some research has also been done in the Soviet Union, but it is very different. It has emphasized *reflexive control*: programming the computer to be disobedient. The first such programmer was Lefevr, in 1967. In 1969 Baranov and Trudolyubov extended his work by making the computer win this disobedience game:



The human begins by choosing either node 9 or node 26, *but doesn't tell the computer which node he has chosen.* The computer starts at node 12; on each turn, it moves to an adjacent node. When it reaches either node 9 or node 26, the game is over: if the node it reaches is the one the human chose, the human wins; if it reaches the opposite node, the computer wins. Before each move, the human tells it where to go; but the computer may decide to do the opposite (disobey).

What strategy should the computer use? If it always obeys, or always disobeys, the human will catch on, and will make it lose.

Instead, Baranov and Trudolyubov programmed the computer to react as follows: obey the human twice, then disobey three times, then obey once, disobey thrice, obey once, disobey twice, obey thrice, disobey once, obey thrice, disobey once,... The irregular alternation of obedience and disobedience confuses the human in a way that works to the computer's advantage. Using that strategy, the computer played against 61 humans, and won against 44 of them (72%). In other words, the typical human tried to mislead the computer but in fact "clued it in" to the human's goal.

Later experiments with other games indicated that the following pattern of disobedience is usually more effective:
obey the human twice, disobey thrice, obey once, disobey four times, obey once, disobey thrice, obey thrice, disobey twice, obey thrice, disobey once, obey once, disobey once.

3. MISINFORMATION

The top researchers in artificial intelligence often exaggerate about their achievements, and underestimate the time it will take to develop a truly intelligent computer. Don't believe everything they say.

In 1957 Herbert Simon said that "within ten years a digital computer will be the world's chess champion." In 1967, when the ten years had elapsed, the only decent chess program was Greenblatt's, which the American Chess Federation rated "class D" (the worst tournament class). Improvement since then has been very slight, and it is highly unlikely that the computer will become a chess master in the near future, let alone world champion.

In 1957 Simon also said that "within ten years a digital computer will discover and prove an important new mathematical theorem." In the almost twenty years that have elapsed since then, the computer has neither discovered nor proved any important new mathematical theorem.

In 1958 Newell, Simon, and Shaw wrote a program to play chess, which they admitted was "not fully debugged" so that one "cannot say very much about the behavior of the program"; but they claimed it was "good in spots (opening)". In 1959 the founder of cybernetics, Norbert Wiener, exaggerated about their program; he told the New York University Institute of Philosophy that "chess-playing machines as of now will counter the moves of a master game with the moves recognized as right in the text books, up to some point in the middle game." In the same symposium Michael Scriven carried the exaggeration even further, and said that "machines are already capable of a good game". In fact, the program they were describing played very poorly, and in its last official bout (October 1960) was beaten by a ten-year-old novice.

In 1960 Herbert Gelernter, who wrote the geometry-theorem program, said, "And today, hardly an expert will contest the assertion that machines will be proving interesting theorems in number theory three years hence." Fifteen years have elapsed since then, but neither Gelernter nor anyone else has programmed the computer to prove theorems in number theory.

In June 1963 this article appeared in the *Chicago Tribune*: "The development of a machine that can listen to any conversation and type out the remarks just like an office secretary was announced yesterday by a Cornell University expert on learning machines. The device is expected to be in operation by fall. Frank Rosenblatt, director of Cornell's cognitive systems research, said the machine will be the largest 'thinking' device built to date. Rosenblatt made his announcement at a meeting on learning machines at Northwestern University's Technological Institute." No such machine exists today, let alone in 1963.

Also in 1963, W. Ross Ashby said, "Gelernter's theorem-proving program has discovered a new proof of the *pons asinorum* that demands no construction." He said that the proof is one that "the greatest mathematicians of 2000 years have failed to notice...which would have evoked the highest praise had it occurred." In fact, the *pons asinorum* is just the simple theorem that the opposite angles of an isosceles triangle are equal, and the computer's contructionless proof had already been discovered by Pappus in 300 A.D.

In 1968 the head of artificial intelligence in Great Britain, Donald Michie, said that "today machines can play chess at championship level." In fact, when computers are allowed to participate in human chess tournaments, they almost always lose.

In 1970 the head of artificial intelligence at MIT, Marvin Minsky, said, "In from three to eight years we will have a machine with the general intelligence of an average human being. I mean a machine that will be able to read Shakespeare, grease a car, play office politics, tell a joke, have a fight. At that point, the machine will begin to educate itself with fantastic speed. In a few months it will be at genius level, and a few months after that its powers will be incalculable." His prediction that it would happen in three to eight years—between 1973 and 1978—is ridiculous. I doubt it will happen during this century, if at all.

Exaggerations concern not only the present and future but also the past. Back in 1962 Arthur Samuel's checker program won one game against Robert Nealey, "a former Connecticut checkers champion". Notice that Nealey was *a former* champion, not *the current* champion when the game was played. Also notice the program won a single game, not a match; and in fact it lost to Nealey later. In 1971 James Slagle slid over those niceties, when he just said that the program "once beat the champion of Connecticut." More recent writers, reading Slagle's words, have gone a step further and omitted the word *once*: one textbook says that "the current program beat the champion of Connecticut." It's not true.

Why do the leaders of artificial intelligence consistently exaggerate? The answer is obvious: to get more research funds from the government. Hubert Dreyfus, chairman of the philosophy department at Berkeley, annoys them by attacking their claims.

4. THE BRAIN

Will the computer be able to imitate the human brain? Opinions vary.

Marvin Minsky, head of artificial intelligence at MIT, says *yes*: "After all, the human brain is just a computer that happens to be made out of meat."

Biologists argue *no*: the brain is composed of 12 billion *neurons*, each of which has between 5,600 and 60,000 *dendrites* for input and a similar number of *axons* for output; the neurons act in peculiar ways, and no computer could imitate all that with complete accuracy—"The neuron is qualitatively quite different from 'on-off' components of current computers."

Herbert Simon, head of artificial intelligence at Carnegie and a psychologist, points out that certain aspects of the brain, such as short-term memory, are known to have very limited capacity and ability. He believes the inner workings of the brain are reasonably simple; it produces complicated output only because it receives complicated input from the sense organs and the environment: "A man, viewed as a behaving system, is quite simple. The apparent complexity of his behavior over time is largely a reflection of the complexity of the environment in which he finds himself." Simon believes that if a computer were given good sense organs, the ability to move, and an elementary ability to learn, and were placed in a stimulating environment (unlike the dull four walls of a computer center), it would start acting in complex ways also.

Hubert Dreyfus, chairman of the philosophy department at Berkeley, argues that progress in artificial intelligence has been very small, is being blocked now by impenetrable barriers, and—most important—the computer's approach to solving problems bears little relationship to the more powerful methods used by humans. He is cynical about the claim that an improvement in computer programs represents progress toward understanding the human mind, which is altogether different: "According to this definition, the first man to climb a tree could claim tangible progress toward reaching the moon. Rather than climbing blindly, it is better to look where one is going."

IN THE HOME

Many computer experts believe the typical American home will be computerized soon. There are two ways to computerize a home:

1. Put microcomputers inside traditional household appliances, to make the appliances more intelligent. Such microcomputers can be bought now for only $15. The digital watches and pocket calculators so many people buy contain stripped-down microcomputers, and it won't be long before microcomputers are implanted within larger appliances as well, for more advanced tasks. Each microcomputer is programmed by the manufacturer to run just one program repeatedly. The family never diddles with the program and is probably unaware that the computer even exists, since it is hidden inside other machinery.

2. Put a terminal in the home. The terminal can communicate with huge computer centers via the telephone. The terminal could be used for amusement—computers have already been programmed to play checkers, chess, Startrek (find the Klingons and kill them!), football (you're the quarterback; do you want a long pass, short pass, simple run, tricky run, or punt?), golf (choose your iron), blackjack (the computer is the dealer), shooting gallery (quick! hit the key before the duck moves), and other games that let people take out their aggressions. It can help you balance your checkbook, analyze phone bills, and plan well-balanced meals for the family, taking into account which foods each member of the family enjoys, and which members are trying to lose weight. It can help schoolchildren, by tutoring them on their homework. Programming it can become an amusing hobby. Equipped with paper, it can be used as an ordinary electric typewriter. You can use a program called RUNOFF, which is available on most computers and makes your typing prettier: it changes your spacing so that the right margin looks as straight as the left. You can make the computer do any kind of thinking you desire, provided someone in the computer center has written a program for it, or you've written one yourself. But the main usefulness of a terminal in the home is for communication: you can communicate with your friends by sending a message over the terminal. No need to wait for the postman to deliver a letter: just type the message on your terminal, type the address where you want it sent, and it will be transmitted instantaneously to your friend's terminal, even if it's thousands of miles away. If he's not "in" at the moment, the message will be stored inside the computer; when he turns his terminal on, the computer will tell him there's a message waiting for him, and ask him whether he'd like to see it. Using the terminal that way—as a communications tool—will change our whole style of living. Here are three views on the subject....

COMPUTERS IN THE HOME

by John Kemeny (abridged)

In 1965, I predicted that computer terminals in the home will be widespread by 1990. How feasible does that prediction look in view of what is known about computers today?

Suppose that we set an initial target of bringing computer service to three million homes. Nine regional centers would cover the United States efficiently, keeping the cost of communication within reasonable bounds. Therefore the average center would have to service some 300,000 customers; however, the key question in any time-sharing system is not how many users there are altogether but how many of them wish to use the system at any given time. If we assume that an average home will use its terminal an hour a day, it is reasonable to guess that at peak periods 10 percent of the users may be on simultaneously. Therefore a regional center must be able to handle 30,000 simultaneous users.

The easiest problem to solve is that of the computer memory. Since there are now on the market laser-beam memories that are able to store and retrieve efficiently the equivalent of 1000 sets of the *Encyclopaedia Britannica*, even today's technology is sufficient to solve the memory problem. The assumption that the average home would want a piece of the memory equivalent to ten pages of the *Encyclopaedia Britannica* is a very generous estimate. That means that a 1000-encyclopaedia memory could accomodate over two million users! I cannot see any difficulty in meeting the demand.

Extrapolating from the cost of running Dartmouth's computer center, but assuming that by 1990 we will have computers that are ten times as fast and cost only three times as much, each regional center will require an annual budget of $30 million. That is only $100 per customer per year.

One major bottleneck remains—namely, reducing the cost of terminals. But by 1990, with a market of millions of customers, a reliable and attractive terminal will surely be within the means of the average family.

What kind of services could a computer network provide? Let us consider a system under which the New York Times, instead of publishing hundreds of thousands of copies, would store the same information in a computer tied into a national network, from which each reader could retrieve the items he wanted, in as much detail as he desired. Each reporter would file his stories by typing them directly into a computer terminal.

A reader sitting in his home could dial into his computer network and ask for his personalized New York Times. The computer would have on file those general topics which the user normally reads. For example, he might normally like to see what is new on the international front, in finance, and in sports. The computer would take these topics up one at a time. In the sports section, it would give the reader a choice of all the different sports covered in that day's newspaper. Let us say that the reader selects baseball and tennis. The computer would list all the available stories under baseball. Each story he chooses would then be presented to him a frame at a time, and he could ask for more details or not as he wished.

I would like to summarize the advantages of this scheme. First, each reader has available to him at any moment, day or night, a completely up-to-date newspaper. It should be perfectly practical to have no more than an hour's lag between the occurrence of a news event and its availability in one's personalized newspaper. Second, the reader would select just those items that are of particular interest to him. This will save him a good deal of time and yet assure him that he does not miss any news that concerns him.

It may be objected that while this scheme sounds great from the reader's point of view, it would bankrupt the Times, which depends on its advertising revenue. But this need not be the case. The computer network would charge the user for various services and could pay a royalty to the Times for each access by a user. Given that the cost of print and distribution is eliminated, the Times may make a profit on "sales" alone. Nor is it necessary to eliminate all advertising. Classified ads could be well indexed and made available to any user who desired to see them. If that did not suffice, the newspaper retrieval program could be written so that between frames it printed ads. However, I would then hope that by paying an extra fee I would have the option of eliminating all advertisements.

I make this suggestion freely available to The New York Times. I suggest that if the newspaper adopts it, it change its motto to "All the news that you see fit to read."

My sister says she would love to see home terminals provide easy information on what is available in various stores. She wears rather narrow shoes and often has to go to a dozen different stores before she finds a pair in her size. Businesses could keep their inventory inside the memory of the computer, and any customer in the region could carry out a simple computerized search to find out what is available. The computer would not only record a sale as a financial transaction but would automatically update the inventory of the store. Therefore a clerk or a customer could know immediately what is available, and when certain items are running low the computer could make out an order for replacements.

I am convinced that for items exceeding one dollar in price cash transactions will totally disappear. The multiplicity of credit cards is already a step in that direction. However, popular as today's credit cards are, the system is basically inefficient. You present your credit card in a store and sign a bill. This has to be sent by mail to an office of the credit card company, which then has to bill you and collect a check from you. In turn, that check is sent to your bank which transfers money to the bank of the credit card company which in turn pays the store. There is enormous waste in all this. As a matter of fact, I suspect that one reason credit cards are so popular is that it often takes as much as two months to receive a bill for a purchase.

A more practical system would be for each person to receive a credit card from his own bank. When buying an item you would present your credit card for insertion in an appropriate computer terminal. This would automatically result in a credit to the account of the store and a debit to your account. If during the transaction you exceed your balance, a red light would come on and the sale would be refused. Or, more likely, the system by which banks provide automatic loans will be expanded, so that moderate overdrafts would result in an interest charge rather than in the refusal of payment.

A pleasant feature of this system would be the fact that it will not be necessary to balance one's checkbooks. This would be done automatically by the bank's computer, and the customer could call up on his home terminal at any time to find out what his balance is.

Let me speculate as to how each member of the family might use the computer terminal. Father, if he brings his work home from the office, can use the terminal in place of a sizable office staff. It can have access to the files of his company, it can help him in carrying out research, and at somewhat greater cost it could even transcribe letters and send them for him.

Mother can do most of her shopping through a computer terminal. She can either find out what is available and go in person to make the purchase or actually enter her shopping list in the terminal. The list would be "delivered" to the local supermarket and the goods automatically packaged and loaded on the truck. If by 1990 the roles of man and woman have been completely reversed, the computer terminal will be equally happy to work out business problems for mother and to help father with his shopping and housework.

Children will find the home terminal an immeasurable asset in doing homework. Indeed the child of 1990 will find it impossible to conceive how the older generation managed to get through school without the help of a computer.

All these services can be highly personalized. The computer would be happy to address the user by first name, nickname, or title, once the user's preferences are known. It could store measurements for various items of clothing to aid in computerized shopping. It could also keep a record of recently performed services so that the computer could "learn" its customer's preferences.

All the applications I have mentioned are possible in principle today. Therefore they should be perfectly practical on a large scale by 1990.

Not only do I forsee home terminals as commonly in existence by 1990, but I fully expect such advertisements as "Why walk all the way downstairs to use your friendly home computer? Get an extension for your bedroom!"

EFFECTS ON THE CITY It is worthwhile to analyze the reasons that millions of people rush into central cities each morning. Some of this manpower is necessary to staff the remaining manufacturing installations, but these are rapidly deserting the major cities. Some of it is necessary to maintain the life functions of a city with millions of people. But the major fraction of the daily in-and-out movement is attributable to executives, professionals, and white-collar workers who staff offices located in the cities. Why is it necessary for them to make a daily pilgrimage from their homes to the city?

One reason is the desire to talk with colleagues in the office or to counterparts in other offices. An increasing percentage of this talk is now carried on over the telephone. I have heard executives complain that they rush into their offices and then spend half their time talking on the telephone, something they could have done equally well at home.

I predict a dramatic effect on the pattern of employment and location of offices will come about by the widespread use of videotelephones. I myself prefer a face-to-face conversation to a discussion by telephone. However, I feel that with the existence of video telephones conversations and small group discussions at remote locations will be almost as effective as face-to-face discussions.

A second major reason for going to the office is that that is where the files are. However, such information will in the future be kept in national computer networks. Once that occurs, the files of a company will be accessible and updatable from any location in the country.

There is no reason why each executive could not pick a site for his office near his home and recruit secretaries and assistants from the immediate vicinity. The small offices would be linked by means of the national computer networks and the network of video-phones. Such an arrangement could greatly reduce the ridiculous amount of traveling that Americans take for granted. An eight-hour working day could be eight hours plus five minutes travel in the morning and five minutes travel in the evening rather than the eleven-hour day necessitated by the daily battle of getting to the office and coming home. Human beings would then have much more freedom in deciding where and how they want to live their lives.

It is my conviction that if the need for millions of people to rush in and out of the city every working day is removed, we would be well on our way to a solution of urban problems. What exactly would be the role of the central city? Perhaps it will become truly a center for information where the machines are located but not the human beings who use them. It would still need a substantial staff of supporting technicians to look after the machines and the usual services necessary to support the people living in the city, but the density of population could be substantially decreased. Since cities still would have a central location, easily accessible, they might expand their roles as centers for recreation and entertainment. Cities might become important because of the presence of museums, theaters, concert halls, and sport arenas, and as places of living for those who insist on seeing a play or a sports event in person rather than watching it over television.

HOME SWEET HOME

by James Martin and Adrian Norman (abridged)

The first widespread use of home terminals will probably be sponsored by employers. Some employees should spend at least part of their time working at home rather than traveling to an office. The overhead cost of providing staff with offices and desks is very high, especially in big cities; a company may cut its costs significantly if most of its personnel work at home. Mothers who participate may be relieved of the boredom they feel when they are unable to leave their children.

We see here a beginning of a return to "cottage industry"—a trend that will probably increase greatly during the next few decades. But at the moment a few factors are counteracting this trend. The first is the reluctance of some companies to give their employees home terminals, because such a step seems an encroachment on leisure time. The second is a feeling that, regardless of what they are doing, employees ought to be at their desks from 9 A.M. to 5 P.M. (At home—who knows—they might be watching television!) The third is a feeling that people cannot work at home because the environment is not suitable. We believe that these three views are not generally held by intelligent and enthusiastic members of the community, and that this is an era when we must throw off the mores of tradition and rethink what is applicable to the age of teleprocessing.

Instead of coming into a store, the consumer could scan a list of available goods and their prices at several different shops on the home terminal, and then use the terminal to place the order. An organization selling in this way could cut overheads to a minimum by eliminating stores.

The customer who buys through such an automated catalogue avoids the exasperation of fruitless searches for hard-to-find merchandise. Besides replacement bulbs for projectors condemned by planned obsolescence, spare parts for automobiles or rare phonograph records—all items likely to be found, if at all, in only one store in the city—there are tedious searches to be made for unique items like summer houses for rent, theater tickets for a particular show, or boats with a particular specification. Presently, these are found through "classified ads" in newspapers or through agents. Much of the work of such agents, and of the advertising columns, could be more cheaply and efficiently done by a data bank accessible from terminals.

An automated diary might cost one cent per entry—a small fee to insure that no appointments are missed and no birthdays forgotten.

The telephone line, in addition to its normal use, will probably also be used for activating household appliances. A family driving home after a few days away, will telephone their home and then key some digits on the Touch-Tone phone which switch on the heat or air-conditioning unit. A woman before leaving for work will preprogram her kitchen equipment to cook a meal. She will then phone at the appropriate time and have the meal prepared.

Computer applications will require extensive use of telephone lines to the home. Fortunately, the lines that today connect homes with local telephone exchanges would suffice. In most telephone systems two wires run from the home of each subscriber to the local exchange (central office). With the exception of party lines, these wires are not shared. It is staggering to reflect that in Manhattan, for example, there are two wires under the street for almost every private subscriber. Most of the time the telephone wires are idle, a remarkable waste of an expensive facility. Furthermore, when they are in use these two wires carry less than one-twentieth of the information they could carry.

Without any change in telephone lines, therefore, we could carry out all of the above functions if the appropriate computer's telephone lines were attached to the same local exchange (central office) as ours. There is no technical reason why our telephone bill for *local* computer calls should be much higher than it is today. Long-distance calls to a computer are a different matter, for they do not involve a permanently connected pair of wires. There will be a marked cost advantage to living in a metropolis, near your computer.

THE HOME

by G. Cuttle (abridged)

Anyone who has any doubts about the ability of a computer to cook breakfast has only to remember the state of mind of the average housewife at seven in the morning to realize that preparing breakfast is a very mechanical task indeed. Many other household tasks are equally suitable for computers to invade; indeed, they have already begun to do so in, for example, washing machines and central-heating control systems. At present each piece of equipment needing such a computer has its own small one built in, but the logical development is to have a larger household computer tucked away under the stairs. Circuits could then be wired into the house so that each individual gadget could be plugged in.

The computer under the stairs is, however, only part of the impact modern technology will have on the home. In general, the cost performance of computers improves the larger they are, and thus the facilities we shall be able to obtain by having access to big regional or national machines through terminals in the home should be comparatively effective.

It may be more economic for companies to subsidize home "communications rooms" for their employees than to rent expensive office space to be commuted to. This process has already started in the field of research, and in some establishments such as MIT it is common for computer terminal facilities to be provided in the homes of senior research staff. This is sensible when one considers the tendency for great ideas to materialize in the bath. The future will see such facilities available in less intellectual fields. Many of the better characteristics of the cottage industry may return, particularly in terms of personal freedoms.

Most business and service organizations will handle day-to-day transactions via terminals, so that the householder can examine his bank statement, scan the latest share prices, buy and sell through his stock computer, and lay bets with his bookie computer. Not merely can he obtain theater, concert, and travel tickets through an agency, but the computer facilities will allow him to interrogate the various possibilities, drawing on a bank of information far greater than any agency today could contain in a brochure or even have readily accessible to its own staff. Agencies will compete not so much in the glossiness of their brochures as in the ease with which their clientele can obtain the information they want from the agency computer.

This ability to interrogate also applies in government, both local and national. The householder could readily ask whether any legislation were in progress that affected his neighborhood or interests, and could thus avoid some of the all-too-frequent failures of cummunication between individuals and local government. The individual could have far easier access to his congressman than the present system permits. Conversely, the householder could be interrogated, and this might well be a highly effective means of keeping congressmen in touch with the feelings of their constituents.

I hope the professionalism of doctors and nurses will not blind them to the possibilities in medicine. Automatic diagnosis by computer could be a useful aid. Interrogation through a household terminal, with carefully programmed questions, could pin-point some everyday ailments. Where any doubt existed, communication would be transferred to either the G.P. or, if the broad diagnosis were clear, a specialist, presenting him with a neat summary of the facts elicited by the preliminary questioning. Thus the specialist would be spared the routine probing and the G.P. spared the need to be a near-specialist in everything. The G.P.'s role would, however, remain since in a computer world the personal touch is all the more necessary. Moreover, much treatment can be carried out in the home that today might necessitate hospital treatment. Where observations and tests are required but no emergency is likely to occurr, it may be far cheaper and pleasanter for the patient to have monitoring dquipment brought to his home and connected through the terminal to a hospital computer. The latter would interrogate the equipment at intervals, and any change in the patient's condition would be brought to the attention of hospital staff.

I certainly do not suggest that all these predictions will be fulfilled by 1980, but I suspect that much more will take place by 1990 than most people imagine.

DANGERS

How could computers change human society? The many good ways are obvious. Here are the bad ones.

ERRORS Although the computer can have a mechanical breakdown, the usual reason for computer errors is *mental* breakdown—on the part of the people who run it. The usual computer blooper is caused by a programmer who writes an incorrect program, or a user who inputs a wrong number. If you want the computer to write a check for $10.00, but you forget to type the decimal point, it will nonchalantly write a check for $1000.

The biggest computer blooper ever made occurred at Cape Kennedy. A rocket rose majestically from its launch pad and headed toward Venus. Suddenly it began to wobble. It had to be destroyed after less than 5 minutes of flight. The loss was put at $18,500,000. What went wrong? After much head-scratching, the answer was finally found. In one of the lines of one of the programs, a programmer omitted a hyphen.

In the computer center of a certain city, every inhabitant's vital statistics were put on cards. One lady in the town was 107, but the number 107 wouldn't fit on the card properly, because the space allotted for AGE was only two digits. The computer just examined the last two digits, which were 07, and assumed she was 7 years old. Since she was 7 and not going to school, the computer printed a truant notice. So city officials visited the home of the 107-year-old lady and demanded to see her mother.

A man in Germany received a bill from a computer requesting the payment of "zero deutschmark". He ignored it, but two weeks later the machine sent him a letter reminding him that he had not paid the sum of "zero deutschmark". Two weeks after that another and more strongly worded letter arrived. He still took no action other than photocopying the letters and gleefully showing them to his friends. But the computer persisted and eventually announced that it was referring his failure to pay to company lawyers. So he telephoned the company. They explained to him there was a minor oversight in the program, assured him it was being corrected, but requested him to send a check for "zero deutschmark" to simplify the reconciliation. He duly made out a check for "0.0 DM." and mailed it. Two days later the check was returned to him from the bank with a polite (nonautomated) letter stating that the bank's computer was unable to process the check.

That last anecdote was from Martin and Norman's *The Computerized Society*. This is from Time Magazine:

> Rex Reed, writer and sometime actor, ordered a bed from a Manhattan department store. Three months passed. Then came the long anticipated announcement: the bed will be delivered on Friday. Reed waited all day. No bed. Having disposed of his other bed, he slept on the floor. Next day deliverers brought the bed but could not put it up. No screws. On Monday, men appeared with the screws. But they could not put in the mattresses. No slats. "That's not our department." Reed hired a carpenter to build them; the department store's slats finally arrived 15 weeks later. Undaunted, Reed went to the store to buy sheets. Two men came up and declared: "You're under arrest." Why? "You're using a stolen credit card. Rex Reed is dead." Great confusion. Reed flashed all his identity cards, the detectives apologized—and then tore up his store charge card. Why? "Our computer has been told that you are dead. And we cannot change this."

On a less humorous note, a woman died from freezing, because an errant computer thought she hadn't paid her utility bill.

UNEMPLOYMENT Since the computer is a labor-saving device, it may make laborers unemployed. Clerks and other low-echelon white-collar workers might find themselves jobless and penniless.

The newspaper companies in New York City have realized it would be more economical to hire fewer printers and use computers instead. But the printers union, upset, cried "Breach of contract!" The final solution, acceptable to both the companies and the printers, was to get the computers, hire no new printers, but retain the current ones until retirement.

The advent of computers does not have to mean a net loss of jobs. In fact, new ones are created, as explained on pages 43-44. Not all computer-related jobs require abstract thinking; there is a need for mechanics, typists, secretaries, salesmen, editors, librarians, etc. There is a need for people to tell the programmers what kind of things to program. Running a computer center is a business, and there is a need for businessmen.

When computers do human work, will there be *enough* work left for us humans to do? Don't worry: when no work is necessary, humans have an amazing talent for inventing it. That's the purpose of Madison Avenue—to create new longings. Instead of significantly shortening the work week, Americans have always opted for a work week of nearly equal length but devoted to more luxurious ends. That's the gung-ho protestant work ethic we're so famous for. Computers will change but not reduce our work.

...That's what will happen in the long run. But for the next decade or two, as society shifts to computers, a good many people will be temporarily out of a job.

QUANTIZATION Since the computer handles numbers easily, it encourages people to reduce problems to numbers. That's both good and bad. It's good because it forces people to be precise. It's bad because some people are starting to make quantization a goal in itself, forgetting that it is but a tool to other ends. Counting the words that Shakespeare wrote is of no value in itself: it must be put to some use. In both the humanities and the social sciences, I'm afraid the motto of the future will be, "If you can't think, count." Some cynics have remarked, "The problem with computers is that they make meaningless research possible."

Since only quantifiable problems can be computerized, there is a danger that, in a burst of computer enthusiasm, people will decide that nonquantifiable problems aren't worth investigating, or htat nonquantifiable aspects of an otherwise quantifiable problem should be ignored. John Kemeny gives this example:

> I have heard a story about the design of a new freeway in the City of Los Angeles. At an open hearing a number of voters complained bitterly that the freeway would go right through the midst of a part of the city heavily populated by blacks and would destroy the spirit of community that they had slowly and painfully built up. The voters' arguments were defeated by the simple statement that, according to an excellent computer, the proposed route was the best possible one. Apparently none of them knew enough to ask how the computer had been instructed to evaluate the variety of possible routes. Was it asked only to consider the costs of building and acquisition of property (in which case it would have found routing through a ghetto area highly advantageous), or was it also asked to take into account the amount of human suffering that a given route would cause? Perhaps the voters would even have agreed that it is not possible to measure human suffering in terms of dollars. But if we omit consideration of human suffering, then we are equating its cost to zero, which is certainly the worst of all procedures!

People are being reduced to numbers: telephone numbers, social security numbers, zip codes, etc. When you start treating another human as just a wrong telephone number, and hang up in his face, something is wrong.

ASOCIAL BEHAVIOR The computer is a seductive toy. When you walk up to it, you may expect to spend only a few minutes, but wind up spending several hours instead. Whether catching bugs or playing Startrek, you are likely to while away a good amount of time. You may find yourself spending more time with the computer than with people. That can be dangerous. For the average American child, his mother is a television set. Will the computer terminal replace T.V. as the national fixation?

The computer is easy to get along with—perhaps too easy. It can gripe at you, but it can't yell. If you don't like its behavior, you can always log it off. You can't do the same thing to people. Excessive time spent with the computer can leave you unprepared for the ambiguities and tensions of real life.

The computer replaces warmth by precision. Excessive time spent with it might inhibit your development as a loving individual.

IRRESPONSIBILITY Computerization is part of the oncoming technological beaurocracy. Like all beaurocracy, it encourages the individual to say, "Don't blame me—I can't change the beaurocracy." Only now the words will read, "Don't blame me—the computer did it."

When John Kemeny's sister asked a salesgirl whether a certain item was in stock, the girl said she couldn't answer, because the information was kept by a computer. The girl hadn't been able to answer questions about stock even before the computer came in—the computer was just a new scapegoat.

Computers will eventually be used to run governments and wars. The thought of someone saying, "I can't change that—that's the way the computer does it" is frightening.

CONCENTRATED POWER As computers amass more and more information about people, they will become centers of knowledge. The people who control them—the programmers, sociologists, generals, and politicians—will gain a lot of power. The thought of so much power being concentrated in the hands of a few is frightening. A handful of people, pressing the wrong buttons, could atom-bomb the earth.

Nobody should have complete control over a computer center. The power should be diversified. Sensitive data and programs should be protected by passwords and other devices, so that no single individual can get access to all of it.

CRIME The computer is the biggest tool in the kit of the white-collar criminal. All he has to do is insert a zero on a tape, and the computer will send him a paycheck for ten times the correct amount. To catch such criminals, computers are programmed to do a lot of double-checking; but if the criminal evades the double-checks, he won't be caught. Police have a hard time tracking down computer criminals, because fingerprints and other traditional forms of evidence are irrelevant. Most computers have passwords, to try to stop people from fooling around with sensitive data, but a bright programmer can devise various tricks to get around the passwords. The crudest is to bug the wires that go to the terminals. The cleverest is to slip extra lines into an innocent program, and get someone else to run it; the extra lines transfer money to the programmer's account.

Since you have to be quite smart to be a computer criminal, even if you are caught you are likely to be admired. The usual reaction of people is not—"What a terrible thing you've done!"—but rather—"Gee, you must be smart. Tell me, how did you do it?" A bright button-down computer criminal who steals $100,000 electronically will usually get a lighter sentence than a dude who has to resort to a gun to get $1000. Is that justice?

INVADED PRIVACY Of all the harm computers can do, "invaded privacy" worries people the most. George Orwell, in his book *1984*, warned that someday "Big Brother will be watching you" via a computer. His prediction is already a reality: your whereabouts are constantly checked by computers owned by the FBI, the IRS, the military, credit-card companies, and mail-order houses. My brother once wrote an innocent letter asking for stamps. Instead of using his own name, he used the name of our dog, Rusty. Since then, we've received letters from many organizations, all addressed to "Mr. Rusty". Our dog's name sits in computers all-across the country.

The information computers have stored about you may be misleading. If you never find out about the error, the consequences can haunt you the rest of your life. Examples:

A teacher saw one of the little boyx in her class kiss another boy. She entered on his computerized school records, "displays homosexual tendencies".

According to computer records, a certain man had "three lawsuits against him". In fact, the first was a scare suit 30 years before, over a magazine subscription he had never ordered; the second had been withdrawn after a compromise over a disputed fee; the third case had been settled in his favor.

You should have a right to see what information is stored about you, and change it if it is incorrect. For example, if a teacher or employer writes a "confidential recommendation" about you, you should have a right to examine it. Otherwise, misleading statements about you may haunt you for life. There is a law against slander that is public; we need a law against slander that is "confidential", since "confidential" has come to mean "available to many people but not to the person described".

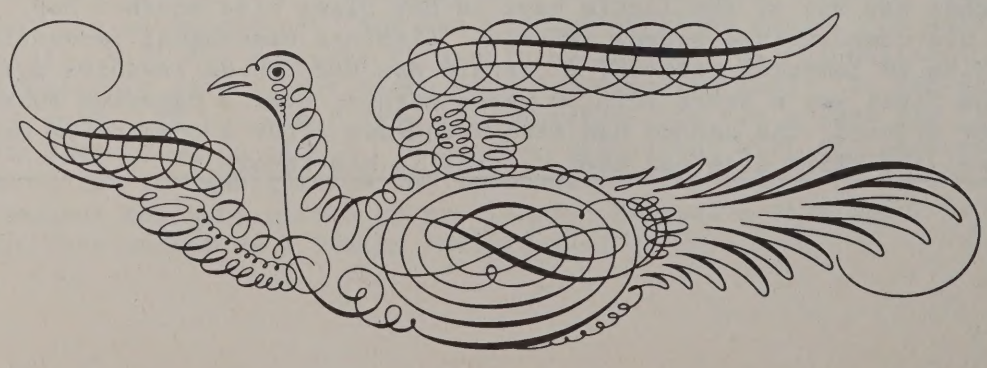
To examine information stored about you, you have to know where it is stored, so many people have proposed this law: if information is stored about an individual in a way that allows the individual to be identified, the organization storing the information must tell the "U.S. Registrar". To find out where information is stored about you, you'd just have to ask the U.S. Registrar. Then you could write to the organizations storing the information, demanding that they tell you what information they stored.

That law would discourage organizations from collecting personal information unnecessarily. Sociologists would be sure to mark on their questionnaires, "Do not sign your name," so the law would not apply to the data they collect.

Even if the information stored about you is accurate, you should have a right to prevent its dissemination to the world in general. Some people have proposed this law: every organization storing information about individuals shall tell the U.S. Registrar what kinds of information are stored, the reason for the storage, and the class of people who are permitted access to the information. The Registrar shall check that no organization stores or disseminates information unjustifiably.

What is "justifiable"? Fearing "Big Brother", people don't want politicians to access personal information. On the other hand, fearing criminals, people want the police to have a free hand in sleuthing. How to give information to the police without giving it to politicians can be puzzling.

Another proposal is that *outdated information should be obliterated.* Many people have suggested such a law, on the grounds that an individual shouldn't be haunted by his distant past; he should be given a chance to turn over a new leaf. Moreover, information that is 50 years old may be couched in words that have been redefined. To be a "leftist", for example, means something different in each decade. It has been proposed that *only facts should be stored, not opinions.* It would be okay to store that someone lives on Fifth Avenue, but not that he lives in a "nice neighborhood".

It's unfortunate that people feel a need for privacy. If the information stored about you is correct, why argue? But many people feel a need to be secretive, and I suppose people do have that right. It is sometimes called the right to be "let alone."

People don't want to feel that their whole life is on stage, recorded by computer. It inhibits them from acting free and natural. Even if the computer doesn't store any damaging information about you, the mere *thought* that your every action is being recorded is damaging, because it makes you act more conservatively. You may be afraid of adopting a good but unusual lifestyle, because anything "different" about you will look bad on the computerized records used by banks, credit-card companies, insurance companies, and other conservative institutions. The harmful thing is not that Big Brother is watching, but that you *feel* he's watching. You are subjugated.